aws

Guide du développeur

Amazon Personalize



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Personalize: Guide du développeur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce qu'Amazon Personalize ? 1
Tarification d'Amazon Personalize 2
Conseils pour les nouveaux utilisateurs 3
À la découverte d'Amazon Personalize avec la Magic Movie Machine 3
Navigation dans les supports de démarrage présentés dans ce guide 3
Relié AWS services et solutions 4
Third-party services 5
En savoir plus 5
Utilisation des AWS SDK 6
Amazon Personalize et l'IA générative 8
Recommandations avec les thèmes du générateur de contenu 8
Métadonnées de recommandation 9
LangChain Code préconfiguré pour la personnalisation 10
Comment ça marche 12
Flux de travail Amazon Personalize 12
Conditions générales d'Amazon Personalize 14
Importation et gestion des données 14
Entraînement 17
Déploiement du modèle et recommandations 20
Amazon Personalize : données 22
Données relatives aux interactions 22
Données relatives à l'article 23
Données de l’utilisateur 23
Données relatives aux actions 23
Données relatives aux actions et aux interactions 24
Configuration d'Amazon Personalize 25
Inscrivez-vous pour un Compte AWS 25
Régions et points de terminaison 25
Configurer les autorisations 25
Autoriser Amazon Personalize à accéder à vos ressources 26
Autoriser les utilisateurs à accéder à Amazon Personalize 30
Permettre à Amazon Personalize d'accéder aux ressources Amazon S3 32
Autoriser Amazon Personalize à utiliser votre AWS KMS clé 39
Configuration du AWS CLI 41

Configuration du AWS Kits SDK 42
Tutoriels de démarrage 44
Conditions préalables pour démarrer 45
Création des données d'entraînement (groupe de jeux de données de domaine) 45
Création des données d'entraînement (groupe de jeux de données personnalisé) 47
Commencer à utiliser un groupe de jeux de données de domaine 47
Commencer à utiliser un groupe de jeux de données de domaine (console) 48
Commencer à utiliser un groupe de jeux de données de domaine (SDK for Java 2.x) 56
Commencer à utiliser un groupe de jeux de données de domaine (SDK pour Python (Boto3)) 65
Commencer à utiliser un groupe de jeux de données de domaine (SDK pour JavaScript v3) 71
Commencer à utiliser un groupe de jeux de données personnalisé 79
Démarrer (console) 80
Démarrer (AWS CLI) 91
Mise en route (SDK pour Python (Boto3)) 102
Mise en route (SDK pour Java 2.x) 108
Adapter votre cas d'utilisation aux ressources Amazon Personalize 121
Cas d'utilisation et fonctionnalités des recettes 122
Personnalisation en temps réel 122
Exploration 123
Mises à jour automatiques 125
Choix d'un cas d'utilisation 127
Cas d'utilisation de VIDEO_ON_DEMAND 127
Cas d'utilisation du commerce électronique 132
Choisir une recette 137
Amazon Personalize les types de recettes par cas d'utilisation 138
Amazon Personalize : recettes 139
Afficher les recettes Amazon Personalize disponibles 141
Personnalisation de l'utilisateur-v2 142
Personnalisation par l'utilisateur 146
Tendance actuelle 164
Comptage de popularité 167
Ranking-V2 personnalisé 168
Personalized-Ranking 172
Similarité sémantique 179

Objets similaires 183
SIMS 187
Meilleure action suivante 195
Affinité entre les objets 200
Affinité entre les attributs et les objets 202
Recettes traditionnelles 205
Préparation des données d'entraînement 227
Directives relatives au format des données en masse pour tous les types de données 229
Données d'interaction entre les articles 230
Exigences relatives aux données d'interaction entre les articles 231
Données d'horodatage 232
Données relatives au type d'événement et à la valeur de l'événement 232
Métadonnées contextuelles 235
Données sur les impressions 236
Exemple de données d'interactions 239
Métadonnées de l'article 241
Exigences relatives aux données relatives aux articles 242
Données d'horodatage de création 244
Métadonnées catégoriques 245
Métadonnées de texte non structurées 246
Données numériques 247
Données de chaîne non catégoriques 248
Exemple de métadonnées d'éléments 248
Métadonnées utilisateur 250
Exigences relatives aux données utilisateur 251
Métadonnées catégoriques 251
Données de chaîne non catégoriques 252
Exemple de métadonnées utilisateur 252
Métadonnées relatives aux actions 253
Exigences relatives aux données relatives aux actions 254
Données d'horodatage d'expiration des actions 255
Répéter les données de fréquence 255
Données de valeur 256
Données d'horodatage de création 256
Métadonnées catégoriques 256
Données de chaîne non catégoriques 257

Exemple de métadonnées d'actions 257
Données d'interaction entre actions 258
Exigences relatives aux données relatives aux interactions entre actions 259
Données relatives au type d'événement 260
Exemple de données sur les interactions entre actions 261
Création de fichiers JSON de schéma 263
Exigences relatives au formatage du schéma 263
Types de données de schéma 264
Ensembles de données et schémas VIDEO_ON_DEMAND 265
Exigences relatives au jeu de données et au schéma du domaine VIDEO_ON_DEMAND ... 266
Exigences relatives au jeu de données sur les interactions 268
Exigences relatives aux ensembles de données utilisateur 270
Exigences relatives à l'ensemble de données 271
Ensembles de données et schémas de commerce électronique 274
Exigences relatives au jeu de données et au schéma du domaine ECOMMERCE 275
Exigences relatives au jeu de données sur les interactions 276
Exigences relatives aux ensembles de données utilisateur 278
Exigences relatives à l'ensemble de données 280
Ensembles de données et schémas personnalisés 283
Exigences relatives aux ensembles de données et aux schémas personnalisés 284
Exigences du schéma du jeu de données d'interactions entre articles 287
Exigences relatives au schéma du jeu de données utilisateur 290
Exigences du schéma du jeu de données d'articles 292
Exigences relatives au schéma du jeu de données Actions 294
Exigences relatives au schéma du jeu de données Action Interactions 297
Liste de vérification du niveau de préparation 299
Avez-vous adapté vos cas d'utilisation aux ressources Amazon Personalize ? 299
Disposez-vous de suffisamment de données sur les interactions entre les articles ? 300
Disposez-vous d'une architecture de diffusion d'événements en temps réel ? 301
Vos données sont-elles optimisées pour Amazon Personalize ? 301
Collectez-vous des données facultatives susceptibles d'améliorer les recommandations ? 302
Avez-vous un plan pour tester vos recommandations ? 302
Avez-vous d'autres objectifs commerciaux ? 302
Création d'un groupe de jeux de données 304
Création d'un groupe de jeux de données (console) 305
Création d'un groupe de jeux de données (AWS CLI) 306

Création d'un groupe de jeux de données (AWS SDKs) 307
Création d'un schéma et d'un jeu de données 310
Création d'un jeu de données et d'un schéma (console) 311
Création d'un jeu de données et d'un schéma (AWS CLI) 312
Création d'un jeu de données et d'un schéma (AWS SDKs) 313
Importation de données d'entraînement 318
Importation de données en masse 319
Modes d'importation 321
Création d'une tâche d'importation de jeux de données (console) 321
Création d'une tâche d'importation de jeux de données (AWS CLI) 324
Création d'une tâche d'importation de jeux de données (AWS SDKs) 326
Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler 330
Informations supplémentaires 332
Configurer les autorisations 333
Lancement de Data Wrangler depuis Amazon Personalize 335
Importation de données dans Data Wrangler 336
Transformation de données 337
Génération de visualisations et d'informations sur les données 339
Traitement des données et importation de celles-ci dans Amazon Personalize 344
Importation d'enregistrements individuels 346
Importer des interactions individuellement 347
Importer des utilisateurs individuellement 350
Importer des articles individuellement 355
Importer des actions individuellement 360
Analyse des données d'entraînement 364
Autorisations requises pour analyser les données 365
Informations sur les données 365
Afficher les informations et les statistiques des jeux de données 369
Recommandeurs de domaines 371
Statuts des recommandeurs 372
Nombre minimal de demandes de recommandation par seconde et auto-scaling 373
Création de recommandeurs de domaines 374
Création de recommandeurs (console) 374
Création d'un recommandeur ()AWS CLI 377
Création d'un recommandeur ()AWS SDKs 377

Activation des métadonnées dans les recommandations 380
Configuration des colonnes utilisées lors de l'entraînement 382
Configuration de l'exploration pour un recommandeur de domaine 384
Évaluation d'un système de recommandation de domaine 388
Récupération des métriques 389
Définitions des métriques 392
Exemple 394
Ressources supplémentaires 394
Mettre à jour un recommandeur 395
Mettre à jour un recommandeur (console Amazon Personalize) 396
Mettre à jour un recommandeur ()AWS CLI 396
Mettre à jour un recommandeur ()AWS SDKs 397
Arrêter un recommandeur 398
Arrêter un recommandeur (console) 399
Arrêter un recommandeur ()AWS CLI 400
Arrêter un recommandeur ()AWS SDKs 400
Ressources personnalisées 403
Configuration d'une solution 404
Création d'une solution 405
Configuration de l'entraînement automatique 414
Configuration des colonnes utilisées lors de l'entraînement 419
Optimisation d'une solution pour un objectif supplémentaire 421
Optimisation d'une solution avec configuration d'événements 429
Hyperparamètres et HPO 434
Choix des données d'interaction entre les objets utilisées pour l'entraînement 438
Clonage d'une solution (console) 440
Mettre à jour une solution 441
Mettre à jour une solution (console) 442
Mettre à jour une solution (AWS CLI) 443
Mettre à jour une solution (AWS SDK) 443
Création manuelle d'une version de solution 444
Création d'une version de solution (console) 445
Création d'une version de solution (AWS CLI) 446
Création d'une version de solution (AWS SDKs) 447
Arrêt de la création d'une version de solution 451
Arrêt de la création d'une version de solution (console) 452

Arrêt de la création d'une version de solution (AWS CLI) 452
Arrêt de la création d'une version de solution (AWS SDKs) 453
Évaluation d'une version de solution 455
Récupération des métriques de version de la solution 456
Définitions des métriques 459
Exemple 466
Ressources supplémentaires 466
Création d'une campagne 467
Mises à jour automatiques des campagnes 468
TPS minimum provisionné 469
Métadonnées des éléments dans les recommandations 470
Création d'une campagne (console) 471
Création d'une campagne (AWS CLI) 473
Création d'une campagne (AWS SDKs) 474
Mettre à jour une campagne 477
Mettre à jour une campagne (console) 478
Mettre à jour une campagne (AWS CLI) 479
Mettre à jour une campagne (AWS SDK) 479
Obtention de recommandations 482
Scores de recommandation 482
Recommandations d'articles en temps réel 483
Comment fonctionne la notation des recommandations (ressources personnalisées) 484
Motifs de recommandation avec User-Personalization-v 2 486
Obtenir des recommandations d'articles en temps réel 486
Obtenir les métadonnées des articles avec des recommandations 491
Articles de promotion 493
Recommandations d'actions en temps réel 504
Comment fonctionne la notation des recommandations d'action 504
Obtenir des recommandations d'action (console) 504
Obtenir des recommandations d'action (AWS CLI) 505
Obtenir des recommandations d'action (AWS SDKs) 506
Obtenir un classement personnalisé (ressources personnalisées) 506
Comment fonctionne le classement personnalisé 507
Obtenir un classement personnalisé (console) 507
Obtenir un classement personnalisé (AWS CLI) 508
Obtenir un classement personnalisé (AWS SDKs) 509

Exemple de carnet ... 513
Utilisation de métadonnées contextuelles ... 513
Obtenir des recommandations à l'aide de métadonnées contextuelles (SDK AWS Python) ... 514
Obtenir des recommandations d'articles par lots ... 515
Flux de travail par lots ... 516
Directives et exigences ... 516
Notation du flux de travail par lots ... 517
Recommandations par lots avec thèmes ... 518
Préparation des données d'entrée pour les recommandations par lots ... 521
Création d'une tâche d'inférence par lots ... 525
Exemples de résultats de tâches d'inférence par lots ... 536
Obtenir des segments d'utilisateurs par lots ... 538
Directives et exigences relatives à l'obtention de segments d'utilisateurs ... 539
Préparation des données d'entrée pour les segments d'utilisateurs ... 540
Création d'une tâche de segmentation par lots ... 543
Exemples de formats de sortie de tâches de segmentation par lots ... 551
Filtrer les résultats ... 552
Expressions de filtrage ... 553
Directives et exigences ... 555
Structure et éléments de l'expression du filtre ... 556
Exemples d'expressions de filtre ... 559
Filtrer les recommandations en temps réel ... 565
Filtrer les recommandations en temps réel (console) ... 565
Filtrer les recommandations en temps réel (AWS CLI) ... 572
Filtrer les recommandations en temps réel (AWS SDK) ... 574
Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées) ... 579
Fournir des valeurs de filtre dans votre JSON d'entrée ... 580
Filtrer les flux de travail par lots (console) ... 581
Filtrer les flux de travail par lots (AWS SDK) ... 582
Enregistrement d'événements ... 583
Comment les événements en temps réel influencent les recommandations ... 584
Enregistrement des événements d'interaction avec un objet ... 585
Exigences relatives à l'enregistrement des événements d'interaction avec les objets et à la formation d'un modèle ... 586
Création d'un outil de suivi des événements liés aux interactions entre objets ... 587

Enregistrement d'un événement d'interaction avec un seul élément 590
Enregistrement d'événements d'interaction avec plusieurs éléments à l'aide des données de valeur des événements 593
Enregistrement des événements d'interaction avec les objets à l'aide des données d'impressions 596
Statistiques relatives aux événements et rapports d'attribution 599
Enregistrement des événements d'interaction entre actions 600
Exigences relatives à l'enregistrement des événements d'interaction entre actions 601
ID de suivi des événements d'action et d'interaction 602
Enregistrement d'un événement d'interaction avec une seule action 602
Enregistrement de plusieurs événements d'interaction entre actions 604
Enregistrement d'événements pour les utilisateurs anonymes 605
Création d'un historique continu des événements pour les utilisateurs anonymes 606
Services tiers de suivi des événements 607
Exemples d’implémentations 607
Maintien de la pertinence des recommandations 609
Maintien à jour des ensembles de données 609
Gestion des recommandeurs de domaines 610
Maintien de solutions personnalisées 610
Mise à jour des données dans les ensembles de données après l'entraînement 612
Comment les nouvelles données influencent les recommandations en temps réel 613
Nouvelles interactions 614
Nouveaux articles 614
Nouveaux utilisateurs 615
Nouvelles actions 616
Comment les nouvelles données influencent les recommandations par lots (ressources personnalisées) 617
Nouvelles interactions 617
Nouveaux utilisateurs 618
Nouveaux articles 618
Remplacement de la structure d'un jeu de données 619
Directives et exigences 619
Remplacement du schéma d'un ensemble de données (console) 621
Remplacement du schéma d'un ensemble de données (AWS CLI) 621
Remplacement du schéma d'un ensemble de données (AWS SDK) 622
Exportation de données 624

Exigences relatives aux autorisations d'exportation de jeux de données 625
Politique de rôle de service pour l'exportation d'un ensemble de données 625
Politique relative aux compartiments Amazon S3 pour l'exportation d'un ensemble de données 626
Création d'une tâche d'exportation de jeux de données dans Amazon Personalize 627
Création d'une tâche d'exportation de jeux de données (console) 627
Création d'une tâche d'exportation de jeux de données (AWS CLI) 629
Création d'une tâche d'exportation de jeux de données (AWS SDK) 630
Supprimer des ressources 634
Directives relatives à la suppression de ressources 634
Ordre recommandé pour la suppression des ressources 636
Suppression des utilisateurs 637
Directives et exigences 637
Préparation d'une liste d'utilisateurs à supprimer 639
Création d'une tâche de suppression de données 639
Suppression d’un jeu de données 643
Supprimer un ensemble de données (console) 644
Supprimer un ensemble de données (AWS CLI) 645
Supprimer un ensemble de données (AWS SDKs) 645
Mesurer l'impact des recommandations 647
Mesurer l'impact des recommandations à l'aide d'une attribution métrique 648
Directives et exigences 649
Création d'une attribution métrique 653
Mettre à jour l'attribution d'une métrique 660
Supprimer une attribution métrique 667
Affichage de graphiques de données métriques dans CloudWatch 669
Publication de rapports d'attribution de métriques sur Amazon S3 670
Mesurer l'impact des recommandations à l'aide A/B de tests 675
Bonnes pratiques en matière de tests A/B 676
Personnalisation des résultats de recherche à partir de OpenSearch 678
Exemple de cas d'utilisation 679
Comment fonctionne le plugin Amazon Personalize Search Ranking 679
Informations supplémentaires 680
Exigences relatives aux plugins 681
Personnalisation des résultats depuis Amazon Service OpenSearch 682
Configurer les autorisations 683

Installation du plug-in 689
Création d'un pipeline 690
Appliquer le plugin 691
Comparaison des résultats 693
Surveillance du plugin 695
Personnalisation des résultats depuis Open Source Open Search 696
Configurer les autorisations 698
Installation manuelle du plugin sur un OpenSearch cluster existant 700
Création d'un nouveau cluster et installation du plugin avec un script 701
Création d'un pipeline 702
Appliquer le plugin 703
Comparaison des résultats 705
Surveillance du plugin 705
Champs de classement de recherche personnalisés 706
Exemple de métriques de pipeline 707
Balisage de ressources 710
Directives et exigences 711
Informations supplémentaires 711
Ajouter des balises aux ressources Amazon Personalize 712
Ajouter des tags (console) 712
Ajouter des tags (AWS CLI) 713
Ajouter des tags (AWS SDKs) 714
Supprimer les tags des ressources Amazon Personalize 717
Supprimer des tags (console) 718
Supprimer les tags (AWS CLI) 718
Supprimer les tags (AWS SDKs) 718
Utilisation de balises dans les politiques IAM 719
Questions fréquentes (FAQ) 721
Importation et gestion des données 721
Création d'une solution personnalisée et d'une version de solution 722
Déploiement de modèles (campagnes personnalisées) 723
Recommandations 724
Recommandations de filtrage 724
Messages d'erreur courants 726
Importation et gestion des données 726
Création d'une solution et d'une version de solution (ressources personnalisées) 727

Déploiement de modèles (campagnes personnalisées) 728
Recommandeurs (groupes de jeux de données de domaine) 728
Recommandations 729
Recommandations de filtrage 729
Spécifier les ressources avec AWS CloudFormation 730
Amazon Personalize et CloudFormation modèles 730
Exemples CloudFormation de modèles pour les ressources Amazon Personalize 731
CreateDatasetGroup 731
CreateDataset 732
CreateSchema 733
CreateSolution 734
En savoir plus sur CloudFormation 734
Exemples de code 736
Amazon Personalize 737
Principes de base 739
Amazon Personalize Events 791
Principes de base 791
Amazon Personalize Runtime 799
Principes de base 800
Sécurité 809
Protection des données 810
Chiffrement des données dans Amazon Personalize 811
Gestion de l’identité et des accès 812
Public ciblé 812
Authentification par des identités 813
Gestion de l’accès à l’aide de politiques 814
Comment Amazon Personalize fonctionne avec IAM 816
Cross-service prévention confuse des adjoints 822
Identity-based exemples de politiques 823
Résolution des problèmes 829
Surveillance avec CloudWatch 831
Utilisation CloudWatch des métriques pour Amazon Personalize 832
Accès aux métriques d'Amazon Personalize 833
Création d'une alarme 834
Application de surveillance 836
CloudWatch statistiques pour Amazon Personalize 836

Journalisation des appels d'API Amazon Personalize avec AWS CloudTrail 840
Amazon Personalize les informations dans CloudTrail 841
Exemple : entrées du fichier journal Amazon Personalize 842
Validation de conformité 843
Résilience 844
Sécurité de l’infrastructure 844
Points de terminaison de VPC (AWS PrivateLink) 845
Création d'un point de terminaison VPC d'interface pour Amazon Personalize 846
Création d'une politique de point de terminaison VPC pour Amazon Personalize 846
Points de terminaison et quotas 848
Amazon Personalize : points de terminaison et régions 848
Conformité d’ 848
Quotas de service 848
Demande d’augmentation de quota 859
Référence des API 860
Actions 860
Amazon Personalize 863
Amazon Personalize Events 1123
Amazon Personalize Runtime 1139
Types de données 1157
Amazon Personalize 1160
Amazon Personalize Events 1330
Amazon Personalize Runtime 1346
Types d'erreurs courants 1352
Paramètres communs 1355
Historique de la documentation 1358
..... mccclxxxii

Qu'est-ce qu'Amazon Personalize ?

Amazon Personalize est un service d'apprentissage automatique entièrement géré qui utilise vos données pour générer des recommandations d'articles pour vos utilisateurs. Il peut également générer des segments d'utilisateurs en fonction de l'affinité des utilisateurs pour certains éléments ou les métadonnées des articles.

Les cas d'utilisation courants sont les suivants :

- Personnalisation d'une application de streaming vidéo : vous pouvez utiliser les ressources Amazon Personalize préconfigurées ou personnalisables pour ajouter plusieurs types de recommandations vidéo personnalisées à votre application de streaming. Par exemple, les meilleurs choix pour vous, More like X et les recommandations vidéo les plus populaires.
- Ajouter des recommandations de produits à une application de commerce électronique : vous pouvez utiliser des ressources Amazon Personalize préconfigurées ou personnalisables pour ajouter plusieurs types de recommandations de produits personnalisées à votre application de vente au détail. Par exemple, les rubriques Recommandé pour vous, Produits fréquemment achetés ensemble et Les clients ayant consulté X ont également consulté les recommandations de produits.
- Ajouter à votre application des recommandations d'action optimales en temps réel : vous pouvez utiliser les ressources personnalisables d'Amazon Personalize pour recommander les actions les plus susceptibles d'être entreprises par vos utilisateurs en fonction de leur comportement. Par exemple, vous pouvez ajouter des recommandations en temps réel pour vous inscrire à votre programme de fidélité, télécharger votre application mobile ou vous inscrire à des e-mails promotionnels.
- Création d'e-mails personnalisés : vous pouvez utiliser les ressources personnalisables Amazon Personalize pour générer des recommandations par lots pour tous les utilisateurs d'une liste d'e-mails. Vous pouvez ensuite utiliser un AWS service ou un service tiers pour envoyer aux utilisateurs des e-mails personnalisés recommandant les articles de votre catalogue.
- Création d'une campagne marketing ciblée — Vous pouvez utiliser Amazon Personalize pour générer des segments d'utilisateurs susceptibles d'interagir avec les articles de votre catalogue. Vous pouvez ensuite utiliser un AWS service ou un service tiers pour créer une campagne marketing ciblée faisant la promotion de différents articles auprès de différents segments d'utilisateurs.
- Personnalisation des résultats de recherche : vous pouvez utiliser les ressources personnalisables Amazon Personalize pour personnaliser les résultats de recherche pour vos utilisateurs. Par

exemple, Amazon Personalize peut reclasser les résultats de recherche que vous avez générés. OpenSearch

Dans la plupart des cas d'utilisation, Amazon Personalize génère des recommandations principalement basées sur les données d'interaction entre les articles. Les données d'interaction entre les articles proviennent de l'interaction de vos utilisateurs avec les articles de votre catalogue. Par exemple, les utilisateurs cliquent sur différents éléments. Les données d'interaction entre vos articles peuvent provenir à la fois de vos enregistrements historiques d'interactions groupées dans un fichier CSV et des événements en temps réel de vos utilisateurs lorsqu'ils interagissent avec votre catalogue. Dans certains cas, Amazon Personalize utilise également des données relatives aux articles et aux utilisateurs, telles que le genre, le prix ou le sexe. Et pour les meilleurs scénarios d'action suivants, il utilise les actions et les données d'interaction entre les actions.

Lorsque vous importez des données en masse, vous pouvez utiliser Amazon SageMaker AI Data Wrangler pour importer des données provenant de plus de 40 sources et les préparer pour Amazon Personalize. Pour plus d’informations, consultez Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler.

Amazon Personalize inclut des opérations d'API pour une personnalisation en temps réel et des opérations par lots pour les recommandations groupées et les segments d'utilisateurs. Vous pouvez démarrer rapidement avec des recommandations optimisées pour les cas d'utilisation pour votre domaine professionnel, ou vous pouvez créer vos propres ressources personnalisées configurables.

Rubriques

- Tarification d'Amazon Personalize
- Conseils pour les nouveaux utilisateurs d'Amazon Personalize
- Relié AWS services et solutions
- Third-party services
- En savoir plus
- Utilisation d'Amazon Personalize avec un AWS Kit SDK

Tarification d'Amazon Personalize

Avec Amazon Personalize, il n'y a pas de frais minimaux ni d'engagement initial. Le niveau AWS gratuit fournit un quota mensuel allant jusqu'à 20 Go de traitement de données par AWS région

disponible, jusqu'à 100 heures de formation par AWS région éligible et jusqu'à 180 000 demandes de recommandation. Le niveau gratuit est valable pendant les deux premiers mois d'utilisation.

Pour obtenir la liste complète des frais et des prix, consultez la section Tarification Amazon Personalize.

Conseils pour les nouveaux utilisateurs d'Amazon Personalize

Si vous utilisez Amazon Personalize pour la première fois, les ressources suivantes peuvent vous aider à démarrer.

Rubriques

- À la découverte d'Amazon Personalize avec la Magic Movie Machine
- Navigation dans les supports de démarrage présentés dans ce guide

À la découverte d'Amazon Personalize avec la Magic Movie Machine

The Magic Movie Machine est une expérience d'apprentissage interactive. Il vous permet de découvrir les fonctionnalités d'Amazon Personalize et d'en savoir plus sur la génération de recommandations. Pour une brève introduction, regardez la vidéo ci-dessous. Essayez ensuite la Magic Movie Machine.

Commencer à utiliser Amazon Personalize

Navigation dans les supports de démarrage présentés dans ce guide

Consultez les sections suivantes pour commencer à utiliser Amazon Personalize. Pour obtenir une liste des fonctionnalités, des exigences et des instructions relatives aux données d'Amazon Personalize, consultezListe de vérification du niveau de préparation.

1. Comment fonctionne Amazon Personalize— Cette section présente le flux de travail Amazon Personalize et vous explique les étapes à suivre pour créer des expériences personnalisées pour vos utilisateurs. Cette section inclut également les termes courants d'Amazon Personalize et leurs définitions. Commencez par cette section pour vous assurer que vous comprenez bien les flux de travail et les termes d'Amazon Personalize avant de commencer à recevoir des recommandations.
2. Configuration d'Amazon Personalize— Dans cette section, vous configurez votre Compte AWS, configurez les autorisations requises pour utiliser Amazon Personalize, et configurez les AWS CLI AWS SDK pour utiliser et gérer Amazon Personalize.

3. Tutoriels de démarrage— Dans cette section, vous pouvez commencer à utiliser Amazon Personalize avec un simple jeu de données vidéo. Suivez ces didacticiels pour acquérir une expérience pratique avec Amazon Personalize. Vous pouvez choisir de commencer avec un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé :
   - Pour commencer à créer un groupe de jeux de données de domaine, suivez Conditions préalables pour démarrer puis lancez les didacticiels dansCommencer à utiliser un groupe de jeux de données de domaine.
   - Pour commencer à utiliser un groupe de jeux de données personnalisé, suivez Conditions préalables pour démarrer puis lancez les didacticiels dansCommencer à utiliser un groupe de jeux de données de domaine.

Après avoir terminé l'exercice de mise en route, vous pouvez commencer à terminer le flux de travail Amazon Personalize. Pour les étapes à suivre et les liens vers la documentation, voirFlux de travail Amazon Personalize.

Relié AWS services et solutions

Amazon Personalize s'intègre parfaitement AWS aux autres services et solutions. Par exemple, vous pouvez effectuer les actions suivantes :

- Utilisez Amazon SageMaker AI Data Wrangler (Data Wrangler) pour importer des données provenant de plus de 40 sources dans un ensemble de données Amazon Personalize. Data Wrangler est une fonctionnalité d'Amazon SageMaker AI Studio qui fournit une solution complète pour importer, préparer, transformer et analyser des données. Pour plus d’informations, consultez Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler.
- AWS Amplify À utiliser pour enregistrer les événements d'interaction avec un objet. Amplify inclut une JavaScript bibliothèque permettant d'enregistrer des événements à partir d'applications clientes Web. Et il inclut une bibliothèque pour enregistrer les événements dans le code du serveur. Pour plus d'informations, consultez la documentation Amplify.
- Automatisez et planifiez les tâches Amazon Personalize avec Maintaining Personalized Experiences with Machine Learning. Cette implémentation de AWS solutions automatise le flux de travail Amazon Personalize, y compris l'importation de données, la formation sur les versions de solutions et les flux de travail par lots.

Third-party services

Amazon Personalize fonctionne bien avec divers services tiers.

- Amplitude — Vous pouvez utiliser Amplitude pour suivre les actions des utilisateurs afin de vous aider à comprendre le comportement de vos utilisateurs. Pour plus d'informations sur l'utilisation d'Amplitude et d'Amazon Personalize, consultez le billet de blog du réseau de AWS partenaires (APN) suivant : Measuring the Effectiveness of Personalization with Amplitude and Amazon Personalize.
- Braze — Vous pouvez utiliser Braze pour envoyer aux utilisateurs des e-mails personnalisés recommandant des articles de votre catalogue. Braze est une plateforme de messagerie leader sur le marché (e-mail, push, SMS). Pour un atelier expliquant comment intégrer Amazon Personalize et Braze, consultez l'atelier Amazon Personalize.
- Optimizely — Vous pouvez utiliser Optimizely pour effectuer des tests avec les recommandations d' A/B Amazon Personalize.
- Segment — Vous pouvez utiliser Segment pour envoyer vos données à Amazon Personalize. Pour plus d'informations sur l'intégration de Segment à Amazon Personalize, consultez Amazon Personalize Destination.

Pour obtenir la liste complète des partenaires, consultez Amazon Personalize Partners.

En savoir plus

Les ressources suivantes fournissent des informations supplémentaires sur Amazon Personalize :

- Pour une référence rapide qui vous aidera à déterminer si Amazon Personalize correspond à votre cas d'utilisation, consultez l'aide-mémoire Amazon Personalize dans le référentiel d'exemples Amazon Personalize.
- Pour une série de vidéos expliquant comment utiliser Amazon Personalize, consultez la série de vidéos Amazon Personalize Deep Dive disponible sur YouTube.
- Pour des didacticiels approfondis et des exemples de code, consultez le référentiel GitHub amazon-personalize-samples.

Utilisation d'Amazon Personalize avec un AWS Kit SDK

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. Chaque kit SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python v4	AWS Exemples de code SDK pour Python v4
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
AWS SDK pour Rust	AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
AWS SDK pour Swift	AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n’avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Amazon Personalize et l'IA générative

Amazon Personalize fonctionne bien avec l'intelligence artificielle générative (IA générative). Amazon Personalize Content Generator, à l'aide de l'IA générative, peut ajouter des thèmes intéressants aux recommandations par lots pour des articles connexes. Le générateur de contenu est une fonctionnalité d'IA générative gérée par Amazon Personalize.

Vous pouvez également utiliser les recommandations d'Amazon Personalize pour intégrer Amazon Personalize à votre flux de travail d'IA générative et améliorer l'expérience de vos utilisateurs. Par exemple, vous pouvez ajouter des recommandations aux instructions génératives de l'IA afin de créer du contenu marketing adapté aux intérêts de chacun de vos utilisateurs. Vous pouvez également générer des résumés concis du contenu recommandé ou recommander des produits ou du contenu par le biais de robots de discussion.

La vidéo suivante montre comment améliorer les recommandations avec Amazon Personalize et l'IA générative.

Améliorez les recommandations avec Amazon Personalize et Generative AI

Les fonctionnalités Amazon Personalize suivantes utilisent l'IA générative ou peuvent vous aider à créer des solutions d'IA générative qui créent du contenu personnalisé. Pour des exemples de blocs-notes Jupyter expliquant comment utiliser Amazon Personalize avec l'IA générative, consultez Generative AI with Amazon Personalize dans le référentiel d'exemples Amazon Personalize.

Rubriques

- Recommandations avec les thèmes du générateur de contenu
- Métadonnées de recommandation
- LangChain Code préconfiguré pour la personnalisation

Recommandations avec les thèmes du générateur de contenu

Amazon Personalize Content Generator peut ajouter des thèmes descriptifs aux recommandations par lots. Le générateur de contenu est une fonctionnalité d'IA générative gérée par Amazon Personalize.

Lorsque vous recevez des recommandations par lots avec des thèmes, Amazon Personalize Content Generator ajoute un thème descriptif pour chaque ensemble d'articles similaires. Par exemple, si vous recevez des recommandations d'articles similaires pour un plat pour le petit déjeuner, Amazon

Personalize peut générer un thème tel que Rise and shine ou Morning Essentials. Vous pouvez utiliser le thème pour remplacer un titre de carrousel générique, comme Frequently bought together. Vous pouvez également intégrer le thème dans un e-mail promotionnel ou une campagne marketing pour de nouvelles options de menu.

Pour générer des thèmes, vous importez des données dans les interactions entre articles et les ensembles de données d'articles, vous créez une solution personnalisée avec la recette Similar-Items et vous générez des recommandations par lots. Les données de votre article doivent inclure la description de l'article et les informations relatives au titre. Les descriptions et les titres détaillés des articles aident Content Generator à créer des thèmes plus précis et plus attrayants.

- Pour plus d'informations sur le flux de travail Amazon Personalize, consultezDétails du flux de travail Amazon Personalize.
- Pour plus d'informations sur les recommandations relatives aux lots, consultez Obtenir des recommandations d'articles par lots ouObtenir des segments d'utilisateurs par lots.
- Pour plus d'informations sur la génération de recommandations d'articles avec des thèmes, consultezRecommandations par lots avec des thèmes issus du générateur de contenu.

Métadonnées de recommandation

Lorsque vous recevez des recommandations, Amazon Personalize peut renvoyer les métadonnées relatives à chaque article recommandé depuis votre ensemble de données Articles. Vous pouvez ajouter ces métadonnées, ainsi que les recommandations d'Amazon Personalize, à vos instructions génératives basées sur l'IA afin de générer un contenu plus convaincant.

Par exemple, vous pouvez utiliser l'IA générative pour créer des e-mails marketing. Vous pouvez utiliser les recommandations d'Amazon Personalize et leurs métadonnées, telles que les genres de films, dans le cadre d'une ingénierie rapide pour l'IA générative. Grâce à des instructions personnalisées, vous pouvez utiliser l'IA générative pour créer des e-mails marketing attrayants adaptés aux intérêts de chacun de vos clients.

Pour obtenir des métadonnées de recommandation, vous devez d'abord suivre le flux de travail Amazon Personalize afin d'importer des données et de créer un domaine ou des ressources personnalisées. Lorsque vous créez un outil de recommandation ou une campagne Amazon Personalize, activez l'option permettant d'inclure des métadonnées dans les recommandations. Lorsque vous recevez des recommandations, vous pouvez spécifier les colonnes de données d'article que vous souhaitez inclure.

- Pour plus d'informations sur le flux de travail Amazon Personalize, consultezDétails du flux de travail Amazon Personalize.
- Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultez la section Activation des métadonnées dans les recommandations (ressources du domaine).
- Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultez la section Activation des métadonnées dans les recommandations (ressources personnalisées).
- Pour plus d'informations sur la façon dont vous pouvez utiliser Amazon Personalize avec l'IA générative pour créer des campagnes marketing, consultez Améliorez vos solutions marketing avec Amazon Personalize et l'IA générative.

LangChain Code préconfiguré pour la personnalisation

LangChain est un framework pour développer des applications basées sur des modèles de langage. Il contient du code conçu pour Amazon Personalize. Vous pouvez utiliser ce code pour intégrer les recommandations Amazon Personalize à votre solution d'IA générative.

Par exemple, vous pouvez utiliser le code suivant pour ajouter à votre chaîne les recommandations Amazon Personalize destinées à un utilisateur.

```
from aws_langchain import AmazonPersonalize
from aws_langchain import AmazonPersonalizeChain
from langchain.llms.bedrock import Bedrock

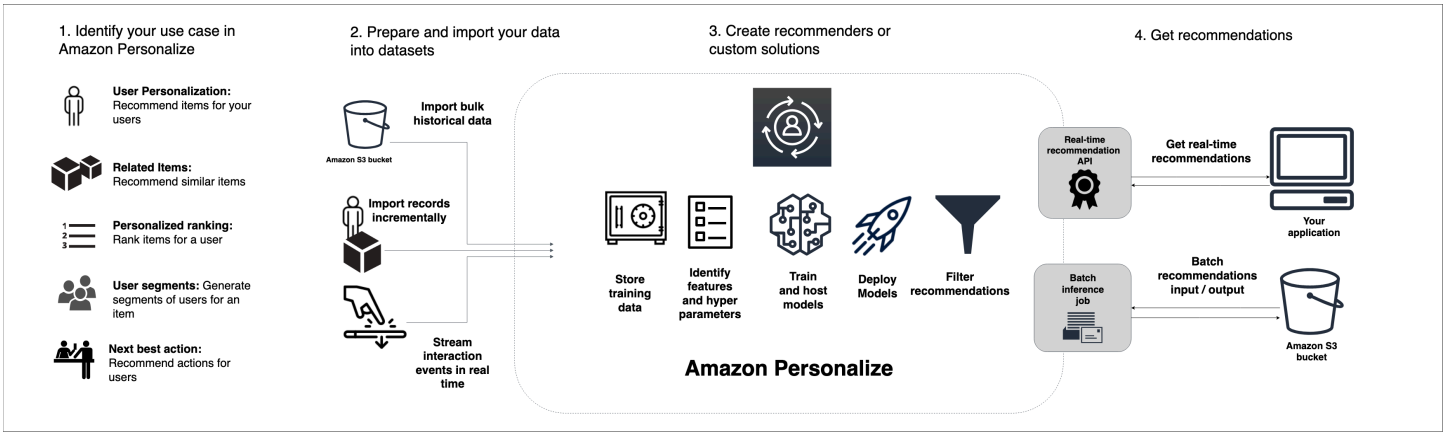
recommender_arn="RECOMMENDER ARN"

bedrock_llm = Bedrock(model_id="anthropic.claude-v2", region_name="us-west-2")
client=AmazonPersonalize(credentials_profile_name="default",region_name="us-
west-2",recommender_arn=recommender_arn)
# Create personalize chain
# Use return_direct=True if you do not want summary
chain = AmazonPersonalizeChain.from_llm(
    llm=bedrock_llm,
    client=client,
    return_direct=False
)
response = chain({'user_id': '1'})
print(response)
```

- Pour plus d'informations sur la mise en route LangChain, consultez l'introduction de la LangChain documentation.
- Pour plus d'informations sur l'utilisation LangChain du code créé pour Amazon Personalize, y compris des exemples de code plus avancés, consultez les LangChain extensions Amazon Personalize dans le référentiel d'AWS exemples.

Comment fonctionne Amazon Personalize

Amazon Personalize utilise vos données pour former des modèles de recommandation personnalisés ou basés sur des domaines. Vous utilisez une API de recommandation privée dans votre application pour demander des recommandations en temps réel. Amazon Personalize prend également en charge les flux de travail par lots, l'obtention de recommandations d'articles et de segments d'utilisateurs.



Rubriques

- Détails du flux de travail Amazon Personalize
- Conditions générales d'Amazon Personalize
- Types de données qu'Amazon Personalize peut utiliser

Détails du flux de travail Amazon Personalize

Le flux de travail Amazon Personalize est le suivant. Pour obtenir une liste des fonctionnalités, des exigences et des instructions relatives aux données d'Amazon Personalize, consultez leListe de vérification du niveau de préparation.

1. Adaptez votre cas d'utilisation aux ressources Amazon Personalize : Amazon Personalize propose des ressources basées sur le domaine et des ressources personnalisées configurées pour différents cas. Lorsque vous associez votre cas d'utilisation à une ressource Amazon Personalize, notez ses exigences en matière de données. Une fois que vous avez choisi un cas d'utilisation ou une recette, ces informations peuvent vous aider à préparer vos données.
2. Préparez vos données d'entraînement : en fonction du cas d'utilisation de votre domaine ou des exigences en matière de données de votre recette personnalisée, préparez vos données

d'entraînement en masse dans un fichier CSV. En fonction de votre cas d'utilisation ou de votre recette, Amazon Personalize peut utiliser les données relatives aux interactions entre les articles, les utilisateurs, les actions et les interactions entre les actions. Si vous ne disposez pas de données en masse, vous pouvez utiliser des opérations d'importation individuelles pour collecter des données et diffuser des événements jusqu'à ce que vous répondiez aux exigences de formation d'Amazon Personalize et aux exigences en matière de données de votre cas d'utilisation ou de votre recette de domaine.

3. Création de fichiers JSON de schéma pour vos données — Créez des fichiers de schéma JSON pour chaque type de données que vous importez. Ces fichiers décrivent la structure et le contenu de vos données, y compris les noms des colonnes et leurs types de données.
4. Créer un groupe de jeux de données — Un groupe de jeux de données est un conteneur pour les ressources Amazon Personalize. Vous pouvez créer un groupe de jeux de données de domaine avec des ressources préconfigurées pour les domaines VIDEO_ON_DEMAND ou ECOMMERCE. Vous pouvez également créer un groupe de jeux de données personnalisé et créer uniquement des ressources personnalisées.
5. Création de schémas et de jeux de données : un schéma indique à Amazon Personalize la structure de vos données et permet à Amazon Personalize d'analyser les données. Un ensemble de données est un conteneur pour les données de formation dans Amazon Personalize.
6. Importer les données d'entraînement dans des ensembles de données : importez vos enregistrements d'interaction, d'élément, d'utilisateur, d'action ou d'interaction d'action préparés. Vous pouvez importer des enregistrements en bloc ou individuellement.
7. Entraînez et déployez un modèle : pour former et déployer un modèle destiné aux domaines VIDEO_ON_DEMAND ou ECOMMERCE, vous devez créer des recommandeurs de domaines. Pour les ressources personnalisées, vous créez une solution personnalisée et une version de solution. Pour obtenir des recommandations en temps réel, vous déployez la version de la solution dans une campagne.
   - Pour plus d'informations sur la création de recommandeurs de domaine, consultezRecommandeurs de domaines.
   - Pour plus d'informations sur la création et le déploiement de ressources personnalisées, consultezRessources personnalisées.
8. Obtenez des recommandations : utilisez votre outil de recommandation ou votre campagne personnalisée pour obtenir des recommandations. Vous pouvez utiliser des filtres pour inclure ou exclure certains types d'éléments des recommandations. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs. Grâce aux ressources

personnalisées, vous pouvez également obtenir des recommandations par lots ou des segments d'utilisateurs sans créer de campagne.

9. Enregistrez les événements en temps réel : enregistrez les événements en temps réel au fur et à mesure que vos clients interagissent avec les recommandations. Cela permet de développer vos données d'interactions et de les actualiser. Il informe également Amazon Personalize des centres d'intérêt actuels de votre utilisateur, ce qui peut améliorer la pertinence des recommandations.

Une fois que vous avez terminé le flux de travail Amazon Personalize pour la première fois, maintenez vos données à jour et réentraînez régulièrement les solutions personnalisées qui utilisent une formation manuelle. Cela permet à votre modèle de tirer des leçons de l'activité la plus récente de vos utilisateurs et de maintenir et d'améliorer la pertinence des recommandations. Pour de plus amples informations, veuillez consulter Maintien de la pertinence des recommandations.

Conditions générales d'Amazon Personalize

Cette section présente les termes utilisés dans Amazon Personalize.

Rubriques

- Importation et gestion des données
- Entraînement
- Déploiement du modèle et recommandations

Importation et gestion des données

Les termes suivants concernent l'importation, l'exportation et le formatage des données dans Amazon Personalize.

jeu de données d'actions

> Un conteneur pour les métadonnées relatives à vos actions. Une action est une activité génératrice d'engagement ou de revenus que vous pouvez recommander à vos utilisateurs, comme l'installation de votre application mobile ou l'adhésion à votre programme de fidélité. Les métadonnées des actions peuvent inclure l'horodatage d'expiration, la valeur, les données de fréquence de répétition et les métadonnées catégorielles de l'action. Ce type de données n'est utilisé que par leNext-Best-Action recette.

jeu de données d'actions et interactions

Un conteneur pour les données historiques et en temps réel que vous collectez à partir des interactions entre les utilisateurs et les actions. Chaque interaction d'action comprend un identifiant utilisateur, un identifiant d'action, un horodatage, un type d'événement et toute donnée supplémentaire concernant l'interaction, telle que des métadonnées catégorielles. Ce type de données n'est utilisé que par leNext-Best-Action recette.

métadonnées contextuelles

Données d'interactions que vous collectez concernant le contexte de navigation d'un utilisateur (tel que l'appareil utilisé ou la localisation) lorsqu'un événement (tel qu'un clic) se produit. Les métadonnées contextuelles peuvent améliorer la pertinence des recommandations pour les utilisateurs nouveaux et existants.

dataset

Un conteneur pour les données que vous chargez sur Amazon Personalize. Il existe cinq types de jeux de données Amazon Personalize : les utilisateurs, les articles, le jeu de données d'interactions entre articles, le jeu de données d'interactions avec les actions et les actions.

groupe de jeux de données

Un conteneur pour les ressources Amazon Personalize, notamment les ensembles de données, les recommandations de domaines et les ressources personnalisées. Un groupe de jeux de données organise vos ressources en collections indépendantes, dans lesquelles les ressources d'un groupe de jeux de données ne peuvent influencer les ressources d'aucun autre groupe de jeux de données. Un groupe de jeux de données peut être un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé.

Groupe de jeux de données de domaine

Un groupe de jeux de données contenant des ressources préconfigurées pour différents domaines commerciaux et différents cas d'utilisation. Amazon Personalize gère le cycle de vie des modèles de formation et leur déploiement. Lorsque vous créez un groupe de jeux de données de domaine, vous choisissez votre domaine professionnel, vous importez vos données et vous créez des recommandations pour chacun de vos cas d'utilisation. Vous utilisez votre outil de recommandation dans votre application pour obtenir des recommandations concernant l' GetRecommendationsopération.

Si vous commencez avec un groupe de jeux de données de domaine, vous pouvez toujours ajouter des ressources personnalisées telles que des solutions et des versions de solutions entraînées avec des recettes pour des cas d'utilisation personnalisés.

Groupe de jeux de données personnalisé

Groupe de jeux de données contenant uniquement des ressources personnalisées, notamment des solutions, des versions de solutions, des filtres, des campagnes et des tâches d'inférence par lots. Vous utilisez une campagne pour obtenir des recommandations concernant l' GetRecommendationsopération. Vous gérez le cycle de vie des modèles de formation et leur déploiement. Si vous commencez avec un groupe de jeux de données personnalisé, vous ne pourrez pas l'associer à un domaine ultérieurement. Créez plutôt un nouveau groupe de jeux de données de domaine.

tâche d'exportation de jeux de données

Outil d'exportation d'enregistrements qui génère les enregistrements d'un ensemble de données vers un ou plusieurs fichiers CSV d'un compartiment Amazon S3. Le fichier CSV en sortie inclut une ligne d'en-tête dont les noms de colonnes correspondent aux champs du schéma du jeu de données.

tâche d'importation de jeux de données

Outil d'importation en masse qui remplit votre ensemble de données Amazon Personalize avec les données d'un fichier CSV dans votre compartiment Amazon S3.

event

Une action de l'utilisateur, telle qu'un clic, un achat ou le visionnage d'une vidéo, que vous enregistrez et importez dans un ensemble de données d'interactions avec Amazon Personalize Item. Vous importez des événements en masse à partir d'un fichier CSV, de manière incrémentielle avec la console Amazon Personalize, et en temps réel.

impressions explicites

Liste des articles que vous ajoutez manuellement à un ensemble de données d'interactions avec Amazon Personalize Item. Contrairement aux impressions implicites, qu'Amazon Personalize obtient automatiquement à partir de vos données de recommandation, vous choisissez ce que vous souhaitez inclure dans les impressions explicites.

impressions implicites

Les recommandations que votre application affiche à un utilisateur. Contrairement aux impressions explicites, que vous ajoutez manuellement à un ensemble de données d'interactions

entre articles, Amazon Personalize extrait automatiquement les impressions implicites à partir de vos données de recommandation.

données sur les impressions

Liste des articles que vous avez présentés à un utilisateur lorsqu'il interagissait avec un article en particulier en cliquant dessus, en le regardant, en l'achetant, etc. Amazon Personalize utilise les données d'impressions pour calculer la pertinence des nouveaux articles pour un utilisateur en fonction de la fréquence à laquelle les utilisateurs ont sélectionné ou ignoré le même article.

jeu de données d'interactions

Conteneur pour les données historiques et en temps réel que vous collectez à partir des interactions entre les utilisateurs et les éléments (appelées événements). Les données d'interactions peuvent inclure des données de type d'événement et des métadonnées contextuelles.

jeu de données d'articles

Un conteneur pour les métadonnées relatives à vos articles, telles que le prix, le genre ou la disponibilité.

fréquence de répétition

Type de métadonnées d'action que vous pouvez importer dans un jeu de données Actions. Les données de fréquence de répétition indiquent le nombre de jours qu'Amazon Personalize doit attendre pour recommander une action particulière après qu'un utilisateur interagisse avec celle-ci, en fonction de l'historique de l'utilisateur dans votre ensemble de données d'interactions avec les actions.

schéma

Un objet JSON au format Apache Avro qui indique à Amazon Personalize la structure de vos données. Amazon Personalize utilise votre schéma pour analyser vos données.

jeu de données des utilisateurs

Un conteneur pour les métadonnées relatives à vos utilisateurs, telles que l'âge, le sexe ou l'adhésion à un programme de fidélité.

Entraînement

Les termes suivants concernent la formation d'un modèle dans Amazon Personalize.

item-to-item recette de similarités (SIMS)

Une recette RELATED_ITEMS qui utilise les données d'un jeu de données Interactions pour recommander des éléments similaires à un élément spécifié. La recette SIMS calcule la similarité en fonction de la façon dont les utilisateurs interagissent avec les articles au lieu de faire correspondre les métadonnées des articles, telles que le prix ou la couleur.

affinité entre les objets

Une recette USER_SEGMENTATION qui utilise les données d'un jeu de données d'interactions entre articles et d'un jeu de données d'articles pour créer des segments d'utilisateurs pour chaque élément que vous spécifiez en fonction de la probabilité que les utilisateurs interagissent avec l'élément.

item-attribute-affinity

Une recette USER_SEGMENTATION qui utilise les données d'un jeu de données d'interactions entre articles et d'un jeu de données d'articles pour créer un segment utilisateur pour chaque attribut d'article que vous spécifiez en fonction de la probabilité que les utilisateurs interagissent avec les éléments dotés de l'attribut.

Next-Best-Action recette

Cette recette génère des recommandations en temps réel pour les prochaines meilleures actions pour vos utilisateurs. La meilleure action suivante pour un utilisateur est celle qu'il est le plus susceptible d'entreprendre. Par exemple, vous inscrire à votre programme de fidélité, télécharger votre application ou demander une carte de crédit. Pour de plus amples informations, veuillez consulter Next-Best-Action recette.

Recette de classement V2 personnalisée

Une recette PERSONALIZED_RANKING qui classe une collection d'articles que vous fournissez en fonction du niveau d'intérêt prévu pour un utilisateur spécifique. Cette recette utilise une architecture basée sur un transformateur pour entraîner un modèle qui apprend à partir des données d'interactions entre les éléments, des métadonnées des éléments et des métadonnées des utilisateurs. Utilisez la recette Personalized-Ranking-v 2 pour personnaliser l'ordre des listes d'éléments ou des résultats de recherche personnalisés pour un utilisateur spécifique. Elle peut s'entraîner sur un maximum de 5 millions d'éléments et générer des recommandations plus pertinentes avec une latence plus faible que la version précédente.

recette à classement personnalisé

Une recette PERSONALIZED_RANKING qui classe une collection d'articles que vous fournissez en fonction du niveau d'intérêt prévu pour un utilisateur spécifique. Utilisez la recette de classement personnalisé pour personnaliser l'ordre des listes d'articles ou des résultats de recherche personnalisés pour un utilisateur spécifique.

recette de comptage de popularité

Une recette USER_PERSONALIZATION qui recommande les éléments ayant le plus d'interactions avec des utilisateurs uniques.

recommandant

Outil de groupe de données de domaine qui génère des recommandations. Vous créez un outil de recommandation pour un groupe de jeux de données de domaine et vous l'utilisez dans votre application pour obtenir des recommandations en temps réel avec l' GetRecommendations API. Lorsque vous créez un recommandeur, vous spécifiez un cas d'utilisation et Amazon Personalize forme les modèles qui soutiennent le recommandeur avec les meilleures configurations pour ce cas d'utilisation.

recipe

Un algorithme Amazon Personalize préconfiguré pour prédire les articles avec lesquels un utilisateur va interagir (pour les recettes USER_PERSONALIZATION), pour calculer des articles similaires à des articles spécifiques qui intéressent un utilisateur (pour les recettes RELATED_ITEMS), ou pour classer une collection d'articles que vous fournissez en fonction de l'intérêt prévu pour un utilisateur spécifique (pour les recettes PERSONALIZED_RANKING).

solution

La recette, les paramètres personnalisés et les modèles entraînés (versions de solution) utilisés par Amazon Personalize pour générer des recommandations.

version de la solution

Un modèle formé que vous créez dans le cadre d'une solution dans Amazon Personalize. Vous déployez une version de solution dans une campagne pour activer l'API de personnalisation que vous utilisez pour demander des recommandations.

mode d'entraînement

L'étendue de la formation à effectuer lors de la création d'une version de solution. Il existe deux modes différents : COMPLET et UPDATE. Le mode FULL crée une toute nouvelle version de solution basée sur l'intégralité des données d'entraînement issues des ensembles de données de

votre groupe de jeux de données. UPDATE met à jour progressivement la version existante de la solution pour recommander les nouveaux éléments que vous avez ajoutés depuis la dernière formation.

> **Note**
>
> Avec User-Personalization-v 2, User-Personalization, ou Next-Best-Action, Amazon Personalize met automatiquement à jour la dernière version de la solution entraînée avec le mode de formation COMPLET. Consultez Mises à jour automatiques.

Recette de personnalisation par l'utilisateur V2

Une recette USER_PERSONALIZATION qui recommande les éléments avec lesquels un utilisateur va interagir en fonction de ses préférences. Cette recette utilise une architecture basée sur un transformateur pour entraîner un modèle qui apprend à partir des données d'interactions entre les éléments, des métadonnées des éléments et des métadonnées des utilisateurs. Elle peut s'entraîner sur un maximum de 5 millions d'éléments et générer des recommandations plus pertinentes avec une latence plus faible que la version précédente.

Recette de personnalisation par l'utilisateur

Une recette USER_PERSONALIZATION basée sur un réseau neuronal récurrent hiérarchique (HRNN) qui prédit les éléments avec lesquels un utilisateur va interagir. La recette de personnalisation par l'utilisateur peut utiliser les données d'exploration des articles et d'impressions pour générer des recommandations pour de nouveaux articles.

Déploiement du modèle et recommandations

Les termes suivants concernent le déploiement et l'utilisation d'un modèle pour générer des recommandations.

période d'optimisation des actions

Période utilisée par Amazon Personalize pour prévoir les actions les plus susceptibles d'être entreprises par l'utilisateur. Par exemple, si la période d'optimisation des actions est de 14 jours, Amazon Personalize prédit les actions les plus susceptibles d'être entreprises par les utilisateurs au cours des 14 prochains jours. Vous configurez la période d'optimisation des actions lorsque vous créez une solution avec leNext-Best-Action recette.

tâche d'inférence par lots

Outil qui importe vos données d'entrée par lots depuis un compartiment Amazon S3, utilise la version de votre solution pour générer des recommandations et exporte les recommandations vers un compartiment Amazon S3. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez une tâche d'inférence par lots pour obtenir des recommandations pour les ensembles de données volumineux qui ne nécessitent pas de mises à jour en temps réel.

tâche de segmentation par lots

Outil qui importe vos données d'entrée par lots depuis un compartiment Amazon S3, utilise la version de votre solution pour créer des segments d'utilisateurs et exporte les segments d'utilisateurs vers un compartiment Amazon S3. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez un travail de segmentation par lots avec une solution basée sur une recette USER_SEGMENTATION pour créer des segments d'utilisateurs en fonction de la probabilité que l'utilisateur interagisse avec différents articles ou des articles dotés d'attributs d'article différents.

campaign

Une version de solution déployée (modèle entraîné) avec une capacité de transaction dédiée allouée pour créer des recommandations en temps réel pour les utilisateurs de votre application. Après avoir créé une campagne, vous utilisez les opérations de l'`getPersonalizedRanking`API `getRecommendations` or pour obtenir des recommandations.

exploration d'objets

Dans le cadre de l'exploration, les recommandations incluent certains éléments ou actions qui sont généralement moins susceptibles d'être recommandés à l'utilisateur, tels que de nouveaux éléments ou actions, des éléments ou des actions avec peu d'interactions, ou des éléments ou actions moins pertinents pour l'utilisateur en raison de son comportement antérieur.

attribution métrique

Un outil que vous utilisez pour mesurer l'impact des recommandations d'articles. Une attribution de mesures crée des rapports basés sur les interactions entre les articles et les données sur les articles que vous importez, ainsi que sur les mesures que vous spécifiez. Par exemple, la durée totale des films visionnés par les utilisateurs ou le nombre total d'événements liés aux clics.

recommandations

Liste des articles avec lesquels Amazon Personalize prévoit qu'un utilisateur interagira. Selon la recette Amazon Personalize utilisée, les recommandations peuvent être soit une liste d'articles (recettes USER_PERSONALIZATION et recettes RELATED_ITEMS), soit le classement d'une collection d'articles que vous avez fournis (recettes PERSONALIZED_RANKING).

segments d'utilisateurs

Listes des utilisateurs qui, selon Amazon Personalize, interagiront avec votre catalogue. En fonction de la recette USER_SEGMENTATION utilisée, vous créez des segments d'utilisateurs en fonction des éléments (recette Item-Affinity) des métadonnées des éléments (recette). Item-Attribute-Affinity Vous créez des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots.

Types de données qu'Amazon Personalize peut utiliser

Les rubriques suivantes présentent les différents types de données que vous pouvez importer dans Amazon Personalize.

Rubriques

- Données relatives aux interactions
- Données relatives à l'article
- Données de l’utilisateur
- Données relatives aux actions
- Données relatives aux actions et aux interactions

Données relatives aux interactions

Une interaction est un événement que vous enregistrez puis importez sous forme de données d'entraînement. Amazon Personalize génère des recommandations principalement sur la base des données d'interactions. Les données relatives aux interactions peuvent inclure les éléments suivants :

- Données relatives au type et à la valeur de l'événement
- Métadonnées contextuelles
- Données sur les impressions

Vous importez des données d'interactions dans un jeu de données d'interactions entre articles. Pour plus de détails sur les ensembles de données d'interactions entre objets, consultezDonnées d'interaction entre les articles.

Données relatives à l'article

Les métadonnées des articles qu'Amazon Personalize peut utiliser sont les suivantes :

- Des données numériques sur chaque article, telles que son prix.
- Des métadonnées catégoriques relatives à chaque élément, telles que le genre ou la couleur de l'élément.
- Données d'horodatage de création pour chaque élément.
- Des métadonnées de texte non structurées, telles que des descriptions de produits ou des synopsis de films.

Vous importez les métadonnées relatives à vos articles dans un jeu de données d'articles. Pour plus d'informations sur les ensembles de données Items, consultezMétadonnées de l'article.

Données de l’utilisateur

Les métadonnées utilisateur qu'Amazon Personalize peut utiliser sont les suivantes :

- Des données numériques sur chaque utilisateur, telles que son âge.
- Des métadonnées catégoriques sur chaque utilisateur, telles que son sexe ou son statut de membre de fidélité.

Vous importez les métadonnées relatives à vos utilisateurs dans un jeu de données Utilisateurs. Pour plus d'informations sur les ensembles de données des utilisateurs, consultezMétadonnées utilisateur.

Données relatives aux actions

Les données d'action qu'Amazon Personalize peut utiliser sont les suivantes :

- La valeur commerciale ou l'importance de chaque action.
- Des métadonnées catégoriques pour chaque action, telles que la saisonnalité ou l'exclusivité de l'action.

- Données d'horodatage d'expiration des actions qui indiquent à quel moment Amazon Personalize doit cesser de recommander chaque action.
- Les données de fréquence de répétition qui indiquent la durée pendant laquelle Amazon Personalize doivent attendre avant de recommander chaque action après qu'un utilisateur interagit avec celle-ci.

Vous importez les données relatives à vos actions dans un jeu de données Actions. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine. Pour plus d'informations sur les ensembles de données Actions, consultezMétadonnées relatives aux actions.

Données relatives aux actions et aux interactions

Les données qu'Amazon Personalize peut utiliser à partir des interactions des utilisateurs avec des actions sont les suivantes :

- Données relatives au type d'événement
- Métadonnées catégoriques

Vous importez des données d'interactions dans un jeu de données d'interactions Action. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine. Pour plus de détails sur les ensembles de données d'interactions Action, consultezDonnées d'interaction entre actions.

Configuration d'Amazon Personalize

Avant d'utiliser Amazon Personalize, vous devez disposer d'un compte Amazon Web Services (AWS) auprès d'un utilisateur administratif. Après avoir configuré les autorisations requises, vous pouvez accéder à Amazon Personalize via la console Amazon Personalize, le AWS Command Line Interface (AWS CLI) ou les AWS SDK.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Régions et points de terminaison
- Configurer les autorisations
- Configuration du AWS CLI
- Configuration du AWS Kits SDK

Inscrivez-vous pour un Compte AWS

Pour commencer AWS, vous avez besoin d'un Compte AWS. Pour plus d'informations sur la création d'un Compte AWS, voir Getting started with an Compte AWS dans le Guide de Gestion de compte AWS référence.

Régions et points de terminaison

Un point de terminaison est une URL qui est le point d'entrée d'un service Web. Chaque point de terminaison est associé à une région AWS spécifique. Faites attention aux régions par défaut de la console Amazon Personalize AWS CLI, des SDK et Amazon Personalize, car tous les composants Amazon Personalize d'une campagne donnée (ensemble de données, solution, campagne, outil de suivi des événements) doivent être créés dans la même région. Pour connaître les régions et les points de terminaison pris en charge par Amazon Personalize, consultez la section Régions et points de terminaison.

Configurer les autorisations

Vous devez autoriser les utilisateurs, les groupes ou les rôles à interagir avec les ressources Amazon Personalize. Vous devez également autoriser Amazon Personalize à accéder aux ressources que vous créez dans Amazon Personalize et à effectuer des tâches en votre nom.

Pour configurer les autorisations

1. Donnez à Amazon Personalize l'autorisation d'accéder à vos ressources dans Amazon Personalize et l'autorisation d'effectuer des tâches en votre nom. Consultez Autoriser Amazon Personalize à accéder à vos ressources.
2. Donnez à vos utilisateurs, groupes ou rôles l'autorisation d'interagir avec les ressources Amazon Personalize et transmettez votre rôle de service à Amazon Personalize. Consultez Autoriser les utilisateurs à accéder à Amazon Personalize.
3. Modifiez la politique de confiance de votre rôle de service Amazon Personalize afin d'éviter le problème de confusion lié aux adjoints. Pour un exemple de politique de relation de confiance, voirCross-service prévention confuse des adjoints. Pour plus d'informations sur la modification de la politique de confiance d'un rôle, consultez la section Modification d'un rôle.
4. Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.
5. Suivez les étapes décrites Permettre à Amazon Personalize d'accéder aux ressources Amazon S3 pour utiliser les politiques relatives aux compartiments IAM et Amazon S3 afin de permettre à Amazon Personalize d'accéder à vos ressources Amazon S3.

Rubriques

- Autoriser Amazon Personalize à accéder à vos ressources
- Autoriser les utilisateurs à accéder à Amazon Personalize
- Permettre à Amazon Personalize d'accéder aux ressources Amazon S3
- Autoriser Amazon Personalize à utiliser votre AWS KMS clé

Autoriser Amazon Personalize à accéder à vos ressources

Pour autoriser Amazon Personalize à accéder à vos ressources, vous devez créer une politique IAM qui fournit à Amazon Personalize un accès complet à vos ressources Amazon Personalize. Vous pouvez également utiliser la `AmazonPersonalizeFullAccess` politique AWS gérée. `AmazonPersonalizeFullAccess`fournit plus d'autorisations que nécessaire. Nous vous recommandons de créer une nouvelle politique IAM qui n'accorde que les autorisations nécessaires. Pour en savoir plus sur les politiques gérées, consultez AWS stratégies gérées.

Après avoir créé une politique, vous créez un rôle IAM pour Amazon Personalize et vous y associez la nouvelle politique.

Rubriques

- Création d'une nouvelle politique IAM pour Amazon Personalize
- Création d'un rôle IAM pour Amazon Personalize

Création d'une nouvelle politique IAM pour Amazon Personalize

Créez une politique IAM qui fournit à Amazon Personalize un accès complet à vos ressources Amazon Personalize.

Pour utiliser l’éditeur de politique JSON afin de créer une politique

1. Connectez-vous à la console IAM Console de gestion AWS et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).

   Si vous sélectionnez Politiques pour la première fois, la page Bienvenue dans les politiques gérées s’affiche. Sélectionnez Mise en route.
3. En haut de la page, sélectionnez Créer une politique.
4. Dans la section Éditeur de politique, choisissez l’option JSON.
5. Entrez le document de politique JSON suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:*"
            ],
            "Resource": "*"
        }
    ]
}
```

6. Choisissez Suivant.

> **Note**
>
> Vous pouvez basculer à tout moment entre les options des éditeurs visuel et JSON. Toutefois, si vous apportez des modifications ou si vous choisissez Suivant dans l'éditeur visuel, IAM peut restructurer votre politique afin de l'optimiser pour l'éditeur visuel. Pour plus d'informations, consultez la page Restructuration de politique dans le Guide de l'utilisateur IAM.

7. Sur la page Vérifier et créer, saisissez un Nom de politique et une Description (facultative) pour la politique que vous créez. Vérifiez les Autorisations définies dans cette politique pour voir les autorisations accordées par votre politique.
8. Choisissez Create policy (Créer une politique) pour enregistrer votre nouvelle politique.

Création d'un rôle IAM pour Amazon Personalize

Pour utiliser Amazon Personalize, vous devez créer un rôle Gestion des identités et des accès AWS de service pour Amazon Personalize. Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM. Après avoir créé un rôle de service pour Amazon Personalize, accordez au rôle les autorisations supplémentaires répertoriées dans la Autorisations de rôle de service supplémentaires liste si nécessaire.

Pour créer le rôle de service pour Personalize (console IAM)

1. Connectez-vous à la console IAM Console de gestion AWS et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
2. Dans le volet de navigation de la console IAM, sélectionnez Rôles, puis Créer un rôle.
3. Pour Trusted entity (Entité de confiance), choisissez Service AWS.
4. Pour Service ou cas d'utilisation, choisissez Personalize, puis choisissez le cas d'utilisation Personalize.
5. Choisissez Suivant.
6. Choisissez la politique que vous avez créée lors de la procédure précédente.
7. (Facultatif) Définissez une limite d'autorisations. Il s'agit d'une fonctionnalité avancée disponible pour les rôles de service, mais pas les rôles liés à un service.

a. Ouvrez la section Définir une limite des autorisations et choisissez Utiliser une limite des autorisations pour contrôler le nombre maximum d'autorisations de rôle.

   IAM inclut une liste des politiques AWS gérées et gérées par le client dans votre compte.

b. Sélectionnez la politique à utiliser comme limite d'autorisations.

8. Choisissez Suivant.
9. Saisissez un nom de rôle ou un suffixe de nom de rôle pour vous aider à identifier l'objectif du rôle.

> **Important**
>
> Lorsque vous nommez un rôle, notez ce qui suit :
>
> - Les noms de rôles doivent être uniques au sein du Compte AWS vôtre et ne peuvent pas être rendus uniques au cas par cas.
>
>   Par exemple, ne créez pas deux rôles nommés **`PRODROLE`** et **`prodrole`**. Lorsqu'un nom de rôle est utilisé dans une politique ou dans le cadre d'un ARN, le nom de rôle est sensible à la casse. Cependant, lorsqu'un nom de rôle apparaît aux clients dans la console, par exemple lors de la procédure d'ouverture de session, le nom de rôle est insensible à la casse.
> - Vous ne pouvez pas modifier le nom du rôle après sa création, car d'autres entités pourraient y faire référence.

10. (Facultatif) Pour Description, saisissez la description du rôle.
11. (Facultatif) Pour modifier les cas d'utilisation et les autorisations du rôle, dans les sections Étape 1 : sélectionner les entités de confiance ou Étape 2 : ajouter des autorisations, sélectionnez Modifier.
12. (Facultatif) Pour identifier, organiser ou rechercher le rôle, ajoutez des identifications sous forme de paires clé-valeur. Pour plus d'informations sur l'utilisation des balises dans IAM, consultez la section Balises pour les Gestion des identités et des accès AWS ressources dans le Guide de l'utilisateur d'IAM.
13. Passez en revue les informations du rôle, puis choisissez Create role (Créer un rôle).

Après avoir créé un rôle pour Amazon Personalize, vous êtes prêt à lui accorder l'accès à votre compartiment Amazon S3 et à toutes AWS KMS les clés.

Autorisations de rôle de service supplémentaires

Après avoir créé le rôle et lui avoir accordé les autorisations nécessaires pour accéder à vos ressources dans Amazon Personalize, procédez comme suit :

1. Modifiez la politique de confiance de votre rôle de service Amazon Personalize afin d'éviter le problème de confusion lié aux adjoints. Pour un exemple de politique de relation de confiance, voirCross-service prévention confuse des adjoints. Pour plus d'informations sur la modification de la politique de confiance d'un rôle, consultez la section Modification d'un rôle.
2. Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Autoriser les utilisateurs à accéder à Amazon Personalize

Pour permettre à vos utilisateurs d'accéder à Amazon Personalize, vous créez une politique IAM qui leur permet d'accéder à vos ressources Amazon Personalize et de transmettre votre rôle de service à Amazon Personalize. Vous utilisez ensuite cette politique lorsque vous ajoutez des autorisations à vos utilisateurs, groupes ou rôles.

Création d'une nouvelle politique IAM pour vos utilisateurs

Créez une politique IAM qui fournit à Amazon Personalize un accès complet à vos ressources Amazon Personalize et aux `PassRole` autorisations nécessaires pour transmettre votre rôle de service à Amazon Personalize (créé enCréation d'un rôle IAM pour Amazon Personalize).

Pour utiliser l'éditeur de politique JSON afin de créer une politique

1. Connectez-vous au Console de gestion AWS et ouvrez la console IAM à https://console.aws.amazon.com/iam/ l'adresse.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).

   Si vous sélectionnez Politiques pour la première fois, la page Bienvenue dans les politiques gérées s'affiche. Sélectionnez Mise en route.
3. En haut de la page, sélectionnez Créer une politique.
4. Dans la section Éditeur de politique, choisissez l'option JSON.
5. Entrez le document de politique JSON suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::123456789012:role/ServiceRoleName",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "personalize.amazonaws.com"
                }
            }
        }
    ]
}
```

6. Choisissez Suivant.

> **Note**
>
> Vous pouvez basculer à tout moment entre les options des éditeurs visuel et JSON. Toutefois, si vous apportez des modifications ou si vous choisissez Suivant dans l'éditeur visuel, IAM peut restructurer votre politique afin de l'optimiser pour l'éditeur visuel. Pour plus d'informations, consultez la page Restructuration de politique dans le Guide de l'utilisateur IAM.

7. Sur la page Vérifier et créer, saisissez un Nom de politique et une Description (facultative) pour la politique que vous créez. Vérifiez les Autorisations définies dans cette politique pour voir les autorisations accordées par votre politique.
8. Choisissez Create policy (Créer une politique) pour enregistrer votre nouvelle politique.

Pour accorder uniquement les autorisations requises pour effectuer une tâche dans Amazon Personalize, modifiez la politique précédente afin d'inclure uniquement les actions requises pour votre utilisateur. Pour obtenir la liste complète des actions Amazon Personalize, consultez la section Actions, ressources et clés de condition pour Amazon Personalize.

Fournir un accès à Amazon Personalize

Joignez la nouvelle politique IAM lorsque vous accordez des autorisations à vos utilisateurs.

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Permettre à Amazon Personalize d'accéder aux ressources Amazon S3

Pour autoriser Amazon Personalize à accéder à votre compartiment Amazon S3, procédez comme suit :

1. Si ce n'est pas déjà fait, suivez les étapes décrites Configurer les autorisations pour configurer les autorisations afin qu'Amazon Personalize puisse accéder à vos ressources dans Amazon Personalize en votre nom.
2. Associez une politique au rôle de service Amazon Personalize (voirCréation d'un rôle IAM pour Amazon Personalize) qui autorise l'accès à votre compartiment Amazon S3. Pour de plus amples informations, veuillez consulter Associer une politique Amazon S3 à votre rôle de service Amazon Personalize.

3. Associez une politique de compartiment au compartiment Amazon S3 contenant vos fichiers de données afin qu'Amazon Personalize puisse y accéder. Pour de plus amples informations, veuillez consulter Associer une politique d'accès Amazon Personalize à votre compartiment Amazon S3.
4. Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

> **Note**
>
> Amazon Personalize ne communiquant pas avec Amazon Personalize AWS VPCs, il ne peut pas interagir avec les compartiments Amazon S3 qui autorisent uniquement l'accès aux VPC.

Rubriques

- Associer une politique Amazon S3 à votre rôle de service Amazon Personalize
- Associer une politique d'accès Amazon Personalize à votre compartiment Amazon S3

Associer une politique Amazon S3 à votre rôle de service Amazon Personalize

Pour associer une politique Amazon S3 à votre rôle Amazon Personalize, procédez comme suit :

1. Connectez-vous à la console IAM (https://console.aws.amazon.com/iam/).
2. Dans le panneau de navigation, sélectionnez Stratégies, puis Créer une stratégie.
3. Choisissez l'onglet JSON et mettez à jour la stratégie comme suit. Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment. Vous pouvez utiliser la politique suivante pour les tâches d'importation de jeux de données ou de suppression de données. Si vous utilisez un flux de travail par lots ou si vous créez une tâche d'exportation de jeux de données, Amazon Personalize a besoin d'autorisations supplémentaires. Voir Politique de rôle de service pour les flux de travail par lots ou Politique relative aux compartiments Amazon S3 pour l'exportation d'un ensemble de données.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

4. Choisissez Suivant : Balises. Ajoutez éventuellement des balises et choisissez Réviser.
5. Donnez un nom à la politique.
6. (Facultatif) Dans Description, entrez une courte phrase décrivant cette politique, par exemple, **`Allow Amazon Personalize to access its Amazon S3 bucket.`**
7. Choisissez Create Policy (Créer une politique).
8. Dans le volet de navigation, choisissez Roles, puis choisissez le rôle que vous avez créé pour Amazon Personalize. Consultez Création d'un rôle IAM pour Amazon Personalize.
9. Pour Autorisations, choisissez Joindre des politiques.
10. Pour afficher la stratégie dans la liste, tapez une partie du nom de la stratégie dans le filtre de requête Filter policies (Filtrer les stratégies).
11. Cochez la case à côté de la politique que vous avez créée plus tôt dans cette procédure.
12. Choisissez Attach policy (Attacher la politique).

Avant que votre rôle ne soit prêt à être utilisé avec Amazon Personalize, vous devez également associer une politique de compartiment au compartiment Amazon S3 contenant vos données. Consultez Associer une politique d'accès Amazon Personalize à votre compartiment Amazon S3.

Politique de rôle de service pour les flux de travail par lots

Pour terminer un flux de travail par lots, Amazon Personalize a besoin d'une autorisation pour accéder à votre compartiment Amazon S3 et y ajouter des fichiers. Suivez les étapes ci-dessus pour associer la politique suivante à votre rôle Amazon Personalize. Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment. Pour plus d'informations sur les flux de travail par lots, consultez Obtenir des recommandations d'articles par lots ouObtenir des segments d'utilisateurs par lots.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Politique de rôle de service pour l'exportation d'un ensemble de données

Pour exporter un ensemble de données, votre rôle de service Amazon Personalize doit être autorisé à utiliser les `ListBucket` actions `PutObject` et sur votre compartiment Amazon S3. L'exemple de politique suivant accorde des `ListBucket` autorisations à Amazon Personalize`PutObject`. `amzn-s3-demo-bucket`Remplacez-le par le nom de votre compartiment et associez la politique à votre rôle de service pour Amazon Personalize. Pour plus d'informations sur l'attachement de politiques à un rôle de service, consultezAssocier une politique Amazon S3 à votre rôle de service Amazon Personalize.

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Associer une politique d'accès Amazon Personalize à votre compartiment Amazon S3

Amazon Personalize a besoin d'une autorisation pour accéder au compartiment S3. Vous pouvez utiliser la politique suivante pour les tâches d'importation de jeux de données ou de suppression de données. Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment. Pour les flux de travail par lots, voirPolitique relative aux compartiments Amazon S3 pour les flux de travail par lots.

Pour plus d'informations sur les politiques relatives aux compartiments Amazon S3, consultez Comment ajouter une politique relative aux compartiments S3 ? .

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
```

```
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Politique relative aux compartiments Amazon S3 pour les flux de travail par lots

Pour les flux de travail par lots, Amazon Personalize a besoin d'une autorisation pour accéder à votre compartiment Amazon S3 et y ajouter des fichiers. Attachez la stratégie suivante à votre compartiment : Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment.

Pour plus d'informations sur l'ajout d'une politique de compartiment Amazon S3 à un compartiment, consultez Comment ajouter une politique de compartiment S3 ? . Pour plus d'informations sur les flux de travail par lots, consultez Obtenir des recommandations d'articles par lots ouObtenir des segments d'utilisateurs par lots.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Politique relative aux compartiments Amazon S3 pour l'exportation d'un ensemble de données

Pour exporter un ensemble de données, Amazon Personalize doit être autorisé à utiliser les `ListBucket` actions `PutObject` et sur votre compartiment Amazon S3. L'exemple de politique suivant accorde le principe `PutObject` et les `ListBucket` autorisations Amazon Personalize. `amzn-s3-demo-bucket`Remplacez-le par le nom de votre compartiment et attachez la politique à votre compartiment. Pour plus d'informations sur l'ajout d'une politique de compartiment Amazon S3 à un compartiment, consultez Comment ajouter une politique de compartiment S3 ? dans le guide de l'utilisateur d'Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Autoriser Amazon Personalize à utiliser votre AWS KMS clé

Si vous spécifiez une clé AWS Key Management Service (AWS KMS) lorsque vous utilisez la console Amazon Personalize ou si vous utilisez votre AWS KMS clé pour chiffrer un compartiment Amazon S3, vous devez autoriser Amazon Personalize à utiliser votre clé. APIs Pour accorder des autorisations, votre politique AWS KMS clé et votre politique IAM associées à votre rôle de service doivent autoriser Amazon Personalize à utiliser votre clé. Cela s'applique à la création des éléments suivants dans Amazon Personalize.

- Groupes de jeux de données
- Tâche d'importation de jeux de données (seule la politique AWS KMS clé doit accorder des autorisations)
- Tâches d'exportation de jeux
- Tâches d’inférence par lots
- Tâches liées au segment Batch
- Attributions métriques

Votre politique AWS KMS clé et vos politiques IAM doivent accorder des autorisations pour les actions suivantes :

- Decrypt
- GenerateDataKey
- DescribeKey
- CreateGrant (obligatoire uniquement dans la politique clé)
- ListGrants

La révocation d'autorisations AWS KMS clés après la création d'une ressource peut entraîner des problèmes lors de la création d'un filtre ou de l'obtention de recommandations. Pour plus d'informations sur AWS KMS les politiques, consultez la section Utilisation des politiques clés dans AWS KMS dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur la création d'une stratégie IAM, consultez la section Création de politiques IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'attachement d'une politique IAM à un rôle, consultez la section Ajouter et supprimer des autorisations d'identité IAM dans le guide de l'utilisateur IAM.

Rubriques

- Exemple de politique clé
- Exemple de politique IAM

Exemple de politique clé

L'exemple de politique clé suivant accorde à Amazon Personalize et à votre rôle les autorisations minimales pour les opérations Amazon Personalize précédentes. Si vous spécifiez une clé lorsque vous créez un groupe de jeux de données et que vous souhaitez exporter des données à partir d'un ensemble de données, votre politique clé doit inclure l'`GenerateDataKeyWithoutPlaintext`action.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-policy-123",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/<personalize-role-name>",
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:ListGrants"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemple de politique IAM

L'exemple de politique IAM suivant accorde à un rôle les AWS KMS autorisations minimales requises pour les opérations Amazon Personalize précédentes. Pour les tâches d'importation de jeux de données, seule la politique AWS KMS clé doit accorder des autorisations.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey",
                "kms:ListGrants"
            ],
            "Resource": "*"
        }
    ]
}
```

Configuration du AWS CLI

The AWS Command Line Interface (AWS CLI) est un outil de développement unifié permettant de gérer des AWS services, notamment Amazon Personalize. Nous vous recommandons de l'installer.

1. Pour l'installer AWS CLI, suivez les instructions de la section Installation du guide AWS Command Line Interface de l'utilisateur de l'AWS Command Line Interface interface.
2. Pour configurer le AWS CLI et configurer un profil pour l'appeler AWS CLI, suivez les instructions de la section Configuration du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.
3. Pour vérifier que le AWS CLI profil est correctement configuré, exécutez la commande suivante.

```
aws configure --profile default
```

Si votre profil a été correctement configuré, vous devez obtenir un résultat similaire à ce qui suit.

```
AWS Access Key ID [****************52FQ]:
AWS Secret Access Key [****************xgyZ]:
Default region name [us-west-2]:
Default output format [json]:
```

4. Pour vérifier que le AWS CLI est configuré pour être utilisé avec Amazon Personalize, exécutez les commandes suivantes.

```
aws personalize help
```

and

```
aws personalize-runtime help
```

and

```
aws personalize-events help
```

Si elle AWS CLI est correctement configurée, vous verrez une liste des AWS CLI commandes prises en charge pour les événements Amazon Personalize, Amazon Personalize Runtime et Amazon Personalize.

Si vous avez configuré le AWS CLI et qu'il ne reconnaît pas les commandes pour Amazon Personalize, mettez à jour le AWS CLI. Pour le mettre à jour AWS CLI, exécutez la commande suivante.

```
pip3 install awscli --upgrade --user
```

Pour plus d'informations, consultez la section Installation du pip AWS CLI d'utilisation.

Configuration du AWS Kits SDK

Téléchargez et installez les AWS SDK que vous souhaitez utiliser. Ce guide fournit des exemples de SDK pour Java 2.x et de SDK JavaScript pour v3. AWS SDK pour Python (Boto3) Pour plus

d'informations sur les autres AWS SDK, consultez la section Outils pour Amazon Web Services. Pour plus d'informations sur la configuration d'Amplify, consultez la documentation Amplify.

- AWS SDK pour Python (Boto3)

  Pour installer le SDK pour Python (Boto3), suivez les instructions de démarrage rapide de la documentation de Boto3.

- SDK pour Java 2.x

  Pour en savoir plus sur la configuration du SDK pour Java 2.x, consultez la rubrique Commencer à utiliser le SDK pour Java 2.x dans le Guide du développeur.AWS SDK for Java 2.x

  Pour des exemples de code pour Amazon Personalize, consultez les exemples de code Java Amazon Personalize dans le référentiel d'exemples de AWS SDK.

- AWS SDK pour JavaScript v3

  Pour en savoir plus sur la configuration du SDK pour la JavaScript version 3, consultez la AWS SDK pour JavaScript rubrique Commencer dans le Guide du AWS SDK pour JavaScript développeur.

  Pour des exemples de code pour Amazon Personalize, consultez les exemples de code Amazon Personalize pour le SDK pour JavaScript v3 dans le référentiel d'exemples de AWS SDK.

Tutoriels de démarrage

Les sections suivantes vous aident à commencer à utiliser Amazon Personalize avec la console Amazon Personalize AWS CLI, et AWS SDKs. Les didacticiels utilisent des données historiques comprenant 100 000 évaluations de films sur 9 700 films provenant de 600 utilisateurs.

Pour simplifier les didacticiels :

- Nous utilisons un petit ensemble de données. Cela peut avoir un impact négatif sur les indicateurs générés par les ressources. Les didacticiels constituent une introduction au flux de travail Amazon Personalize et ne génèrent pas nécessairement les modèles les plus performants.
- Nous créons uniquement un jeu de données sur les interactions entre les éléments et nous nous basons sur le fait qu'un utilisateur a vu un film et non sur l'évaluation qu'il a donnée au film. Cela simplifie la préparation des données de formation.
- Nous n'enregistrons pas les événements d'interaction utilisateur en direct. Pour plus d'informations sur la capture d'événements utilisateur, consultez Enregistrement des événements en temps réel pour influencer les recommandations.

Vous pouvez choisir de commencer avec un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé :

- Les groupes de jeux de données de domaine fournissent des ressources optimisées pour différents cas d'utilisation en fonction de votre domaine. Pour commencer à créer un groupe de jeux de données de domaine, suivez le didacticiel Conditions préalables pour démarrer puis complétez le didacticiel dansCommencer à utiliser un groupe de jeux de données de domaine.
- Les groupes de jeux de données personnalisés vous permettent de créer et de configurer uniquement des ressources personnalisées. Pour commencer à fournir des recommandations de films personnalisées à vos utilisateurs avec des ressources personnalisées et la Recette de personnalisation par l'utilisateur V2 recette, suivez les didacticiels Conditions préalables pour démarrer puis lancez-les dansCommencer à utiliser un groupe de jeux de données personnalisé.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Conditions préalables pour démarrer
- Commencer à utiliser un groupe de jeux de données de domaine
- Commencer à utiliser un groupe de jeux de données personnalisé

Conditions préalables pour démarrer

Les étapes suivantes sont des prérequis pour les exercices de mise en route.

1. Configurez des autorisations pour qu'Amazon Personalize puisse accéder à vos ressources en votre nom. Cela implique de créer un rôle de service pour Amazon Personalize et de lui accorder l'accès aux ressources Amazon Personalize avec une politique IAM. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à accéder à vos ressources.
2. Préparez vos données d'entraînement et chargez-les dans votre compartiment Amazon S3 :
   - Pour les didacticiels sur les groupes de jeux de données de domaine, consultezCréation des données d'entraînement (groupe de jeux de données de domaine).
   - Pour les didacticiels sur les groupes de jeux de données personnalisés, voirCréation des données d'entraînement (groupe de jeux de données personnalisé).
3. Donnez à votre rôle de service Amazon Personalize l'autorisation d'accéder à vos ressources Amazon S3, comme indiqué dansPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Création des données d'entraînement (groupe de jeux de données de domaine)

Pour créer des données d'entraînement, téléchargez, modifiez et enregistrez les données d'évaluation des films dans un bucket Amazon Simple Storage Service (Amazon S3). Donnez ensuite à Amazon Personalize l'autorisation de lire des informations depuis le bucket.

Pour créer les données d'entraînement

1. Téléchargez et décompressez le fichier zip des évaluations des films, ml-latest-small.zip, depuis la MovieLenssection « recommandé pour l'éducation et le développement » (F. Maxwell Harper et Joseph A. Konstan, 2015). Les MovieLens ensembles de données : historique et contexte. Transactions ACM sur les systèmes intelligents interactifs (TII) 5, 4 : 19:1 — 19:19. https://doi.org/10.1145/2827872).

2. Ouvrez le fichier `ratings.csv`. Ce fichier contient les données d'interactions pour ce didacticiel.

   a. Supprimez la colonne rating (évaluation).

   b. Renommez les `movieId` colonnes `userId` et en `USER_ID` et `ITEM_ID` respectivement.

   c. Ajoutez une colonne EVENT_TYPE et définissez la valeur de chaque enregistrement sur. `watch` Si vous utilisez Microsoft Excel, vous pouvez définir le EVENT_TYPE pour chaque enregistrement `watch` en saisissant la première cellule de la colonne, puis en double-cliquant sur le coin inférieur droit de la cellule. Votre en-tête doit être le suivant :

      `USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE`

      Ces colonnes doivent être exactement comme indiqué pour qu'Amazon Personalize reconnaisse les données. Les premières lignes de vos données doivent se présenter comme suit :

      ```
      USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE
      1,1,964982703,watch
      1,3,964981247,watch
      1,6,964982224,watch
      1,47,964983815,watch
      1,50,964982931,watch
      ....
      ....
      ```

   Enregistrez le fichier `ratings.csv`.

3. `ratings.csv`Téléchargez-le dans votre compartiment Amazon S3. Pour plus d'informations, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service.

4. Autorisez Amazon Personalize à lire les données du compartiment. Pour de plus amples informations, veuillez consulter Permettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Création des données d'entraînement (groupe de jeux de données personnalisé)

Pour créer des données d'entraînement, téléchargez, modifiez et enregistrez les données d'évaluation des films dans un bucket Amazon Simple Storage Service (Amazon S3). Donnez ensuite à Amazon Personalize l'autorisation de lire des informations depuis le bucket.

1. Téléchargez et décompressez le fichier zip des évaluations des films, ml-latest-small.zip, depuis la MovieLenssection « recommandé pour l'éducation et le développement » (F. Maxwell Harper et Joseph A. Konstan, 2015). Les MovieLens ensembles de données : historique et contexte. Transactions ACM sur les systèmes intelligents interactifs (TII) 5, 4 : 19:1 — 19:19. https://doi.org/10.1145/2827872).
2. Ouvrez le fichier `ratings.csv`. Ce fichier contient les données d'interactions pour ce didacticiel.

   a. Supprimez la colonne rating (évaluation).

   b. Remplacez la ligne d'en-tête par ce qui suit :

      `USER_ID,ITEM_ID,TIMESTAMP`

      Ces en-têtes doivent être exactement tels qu'ils apparaissent pour qu'Amazon Personalize reconnaisse les données.

   Enregistrez le fichier `ratings.csv`.
3. `ratings.csv`Téléchargez-le dans votre compartiment Amazon S3. Pour plus d'informations, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service.
4. Autorisez Amazon Personalize à lire les données du compartiment. Pour de plus amples informations, veuillez consulter Permettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Commencer à utiliser un groupe de jeux de données de domaine

Dans ce didacticiel de mise en route, vous allez créer un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND, importer des données d'interactions à partir d'un fichier CSV et créer un outil de recommandation avec le cas d'utilisation qui vous convient le mieux.

Ensuite, vous utilisez le logiciel de recommandation pour obtenir des recommandations de films personnalisées pour un utilisateur. Le didacticiel utilise des données historiques comprenant 100 000 évaluations de films sur 9 700 films de 600 utilisateurs.

Pour commencer, complétez le Conditions préalables pour démarrer et, selon la manière dont vous souhaitez créer les ressources Amazon PersonalizeCommencer à utiliser un groupe de jeux de données de domaine (console), passez à Commencer à utiliser un groupe de jeux de données de domaine (SDK pour Python (Boto3))Commencer à utiliser un groupe de jeux de données de domaine (SDK for Java 2.x), ouCommencer à utiliser un groupe de jeux de données de domaine (SDK pour JavaScript v3).

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Commencer à utiliser un groupe de jeux de données de domaine (console)
- Commencer à utiliser un groupe de jeux de données de domaine (SDK for Java 2.x)
- Commencer à utiliser un groupe de jeux de données de domaine (SDK pour Python (Boto3))
- Commencer à utiliser un groupe de jeux de données de domaine (SDK pour JavaScript v3)

Commencer à utiliser un groupe de jeux de données de domaine (console)

Dans cet exercice, vous allez utiliser la console Amazon Personalize pour créer un groupe de jeux de données de domaine et un outil de recommandation qui renvoie des recommandations de films pour un utilisateur donné.

Avant de commencer cet exercice, passez en revue leConditions préalables pour démarrer.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour plus d’informations, consultez Conditions requises pour supprimer des ressources Amazon Personalize.

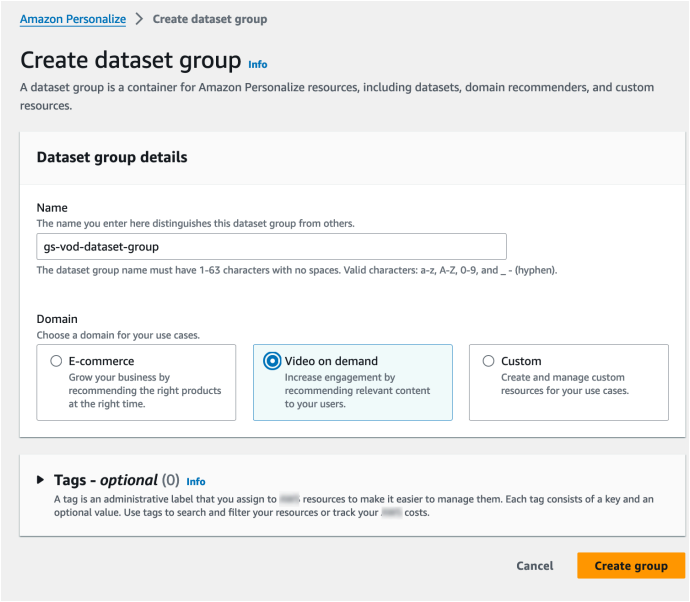
Étape 1 : créer un groupe de jeux de données de domaine

Dans cette procédure, vous créez un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND, vous créez un jeu de données d'interactions entre articles avec le schéma par défaut pour le domaine VIDEO_ON_DEMAND et vous importez les données d'interactions entre

articles dans lesquelles vous les avez créées. Création des données d'entraînement (groupe de jeux de données de domaine)

Pour créer un groupe de jeux de données de domaine

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Dans le volet de navigation, choisissez Create dataset group.
3. Dans Détails du groupe de jeux de données, spécifiez le nom de votre groupe de jeux de données.
4. Dans le champ Domaine, sélectionnez Vidéo à la demande. Le domaine que vous choisissez détermine le schéma par défaut que vous utilisez lors de l'importation de données. Il détermine également les cas d'utilisation disponibles pour les recommandeurs. Votre écran doit se présenter comme suit.



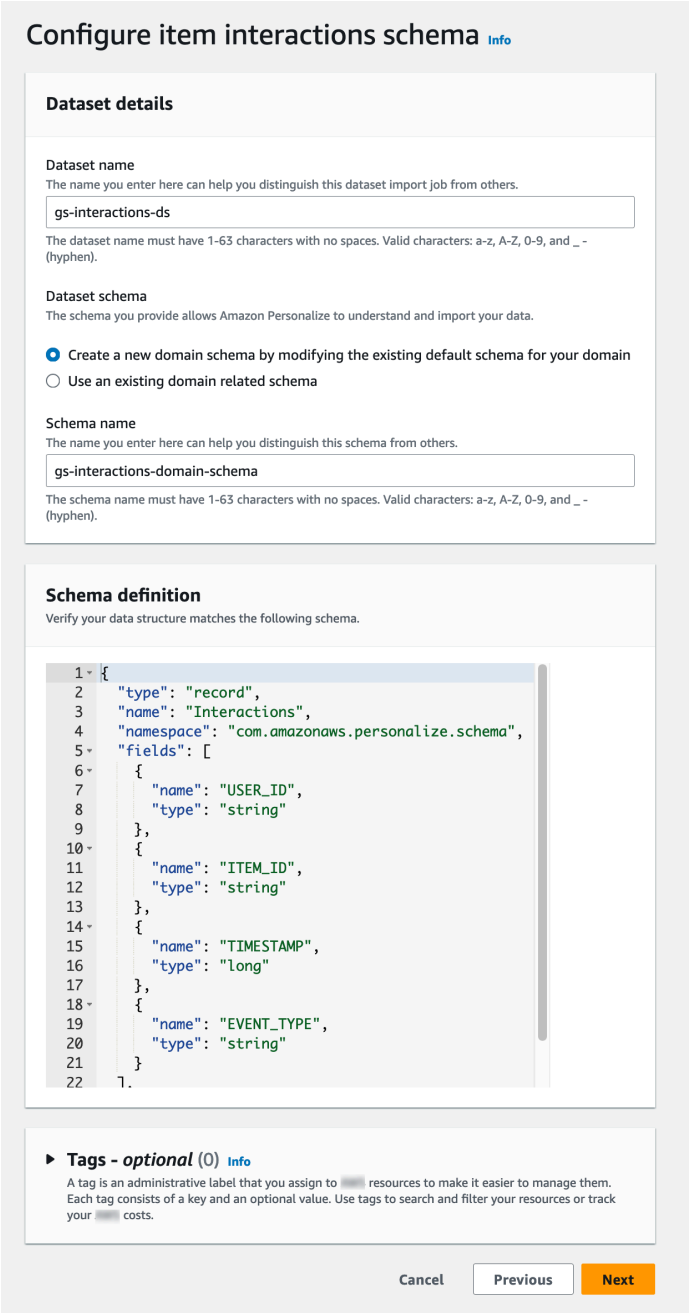
5. Choisissez Create dataset group (Créer un groupe d'ensembles de données). La page d'aperçu apparaît. Passez à Étape 2 : Importer des données.

Étape 2 : Importer des données

Dans cette procédure, vous créez un jeu de données d'interactions entre articles avec le schéma de domaine VIDEO_ON_DEMAND par défaut. Vous importez ensuite les données d'interactions entre les articles que vous avez créées Création des données d'entraînement (groupe de jeux de données de domaine).

Pour importer des données

1. Sur la page Vue d'ensemble, à l'étape 1. Créez des ensembles de données et importez des données, choisissez Create dataset et choisissez Item interactions dataset.
2. Choisissez Importer les données directement dans les ensembles de données Amazon Personalize, puis Next.
3. Sur la page Configurer le schéma des interactions entre articles, pour Nom du jeu de données, saisissez un nom pour votre jeu de données d'interactions entre articles.
4. Pour le schéma de jeu de données, choisissez Créer un nouveau schéma de domaine en modifiant le schéma par défaut existant pour votre domaine et entrez un nom pour le schéma. La définition du schéma est mise à jour pour afficher le schéma par défaut du domaine VIDEO_ON_DEMAND. Laissez le schéma inchangé. Votre écran doit se présenter comme suit.

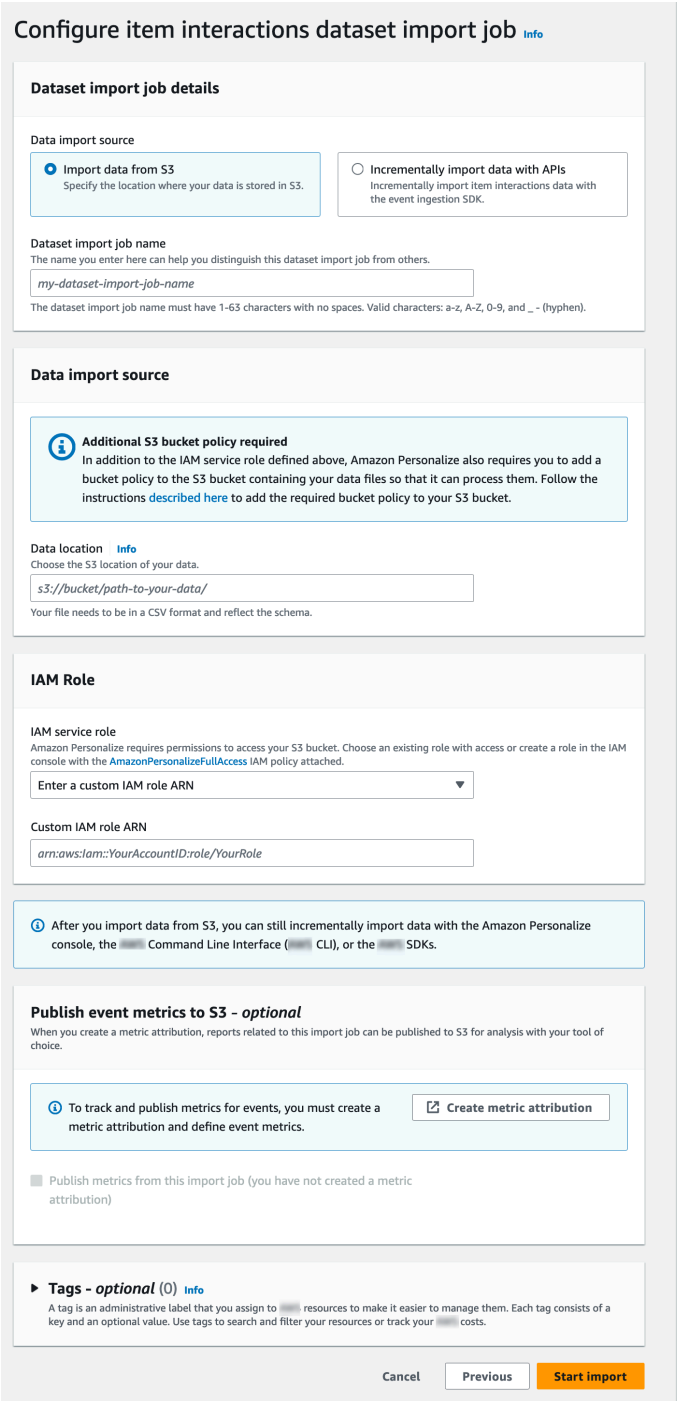


5. Choisissez Suivant. La page de tâche d'importation du jeu de données Configurer les interactions entre les articles apparaît.
6. Sur la page de travail d'importation du jeu de données Configurer les interactions entre les éléments, laissez la source d'importation de données inchangée sous la forme Importer des données depuis S3.
7. Pour le nom de la tâche d'importation du jeu de données, attribuez un nom à votre tâche d'importation.

8. Dans Source d'importation des données, spécifiez où vos données sont stockées dans Amazon Simple Storage Service (S3). Utilisez la syntaxe suivante :

   `s3://amzn-s3-demo-bucket/<folder path>/<CSV filename>`

9. Dans Rôle IAM, pour le rôle de service IAM, choisissez Enter a custom IAM role ARN (ARN) et entrez le nom de ressource Amazon (ARN) du rôle que vous avez créé. Création d'un rôle IAM pour Amazon Personalize Votre écran doit se présenter comme suit.



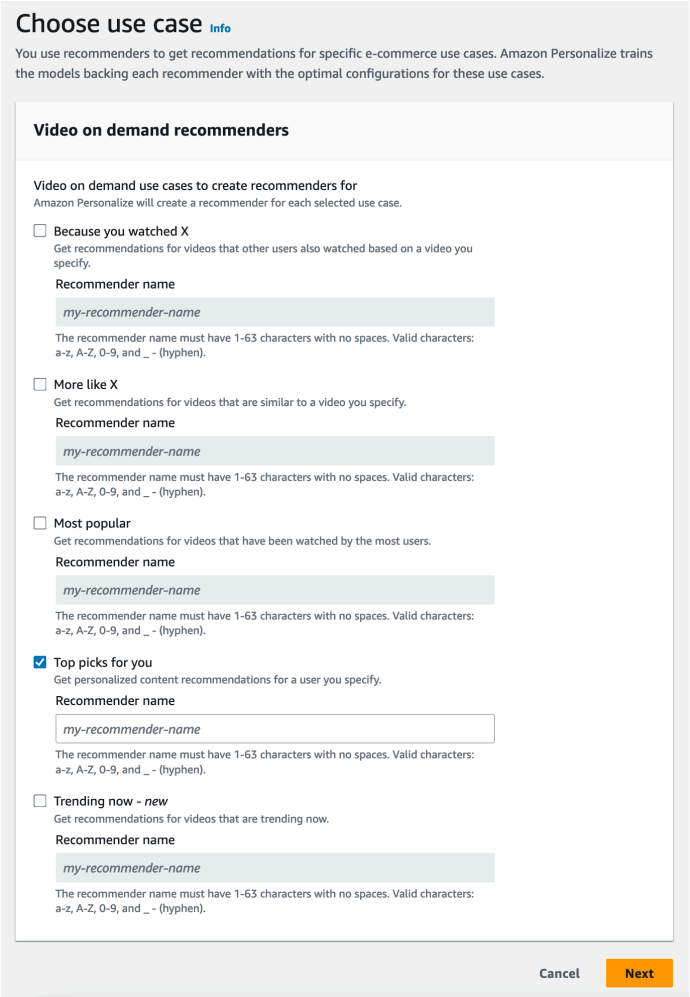
10. Choisissez Démarrer l'importation pour importer des données. La page Vue d'ensemble de votre groupe de données de domaine apparaît. Notez le statut de l'importation dans la section Configurer les ensembles de données. Lorsque le statut est `Interaction data active` passé àÉtape 3 : créer un recommandeur.

Étape 3 : créer un recommandeur

Dans cette procédure, vous créez un outil de recommandation pour le cas d'utilisation Top picks for you pour le domaine VIDEO_ON_DEMAND.

Pour créer un recommendeur

1. Sur la page Vue d'ensemble de votre groupe de données de domaine, à l'étape 3, choisissez l'onglet Utiliser des recommandations de vidéos à la demande, puis sélectionnez Créer des recommandations.
2. Sur la page Choisir un cas d'utilisation, choisissez les meilleurs choix pour vous et indiquez un nom de recommandeur. Votre écran devrait ressembler à ce qui suit.



3. Choisissez Suivant.
4. Laissez les champs de la page de configuration avancée inchangés et choisissez Next.
5. Passez en revue les détails du recommandeur et choisissez Créer des recommandeurs pour créer votre recommandeur.

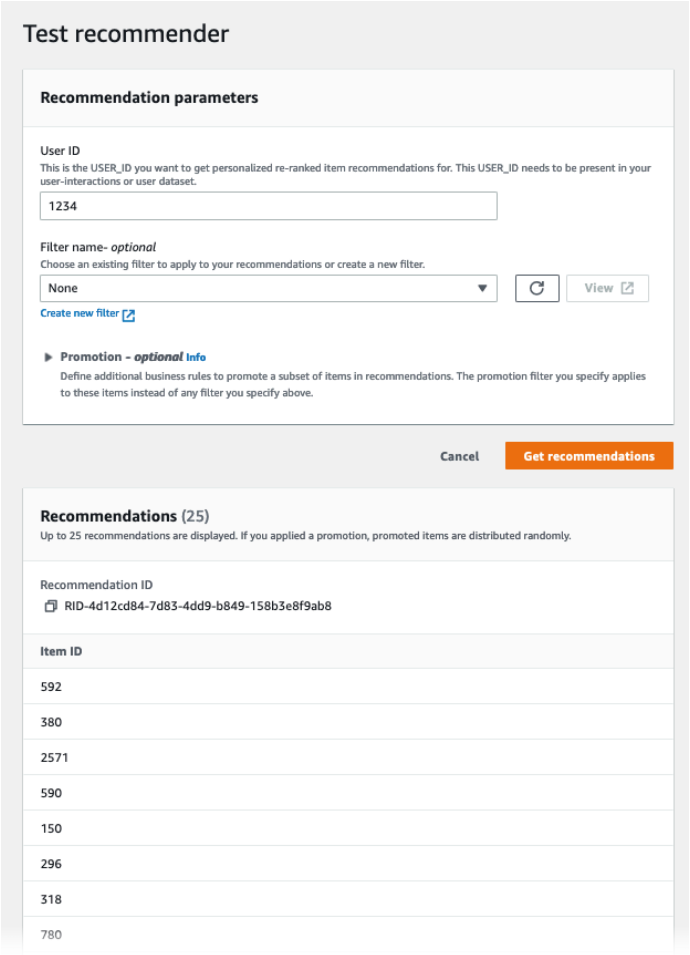
   Vous pouvez suivre le statut de chaque recommandeur sur la page Recommandeurs. Lorsque le statut de votre recommandeur est Actif, vous pouvez l'utiliser pour obtenir des recommandations. Étape 4 : Obtenir des recommandations

Étape 4 : Obtenir des recommandations

Dans cette procédure, vous utilisez le recommandateur que vous avez créé à l'étape précédente pour obtenir des recommandations.

Obtention de recommandations

1. Sur la page Vue d'ensemble de votre groupe de données de domaine, dans le volet de navigation, sélectionnez Recommenders.
2. Sur la page Recommandeurs, choisissez votre recommandeur.
3. En haut à droite, choisissez Test.
4. Dans Paramètres de recommandation, entrez un ID utilisateur. Laissez les autres champs en l'état.
5. Choisissez Obtenir des recommandations. Un tableau contenant les 25 principaux éléments recommandés par l'utilisateur s'affiche. Votre écran doit se présenter comme suit.



Commencer à utiliser un groupe de jeux de données de domaine (SDK for Java 2.x)

Ce didacticiel explique comment utiliser le SDK pour Java 2.x afin de créer un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND. Dans ce didacticiel, vous allez créer un outil de recommandation pour le cas d'utilisation qui vous convient le mieux.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Prérequis

Les étapes suivantes sont nécessaires pour suivre ce didacticiel :

- Complétez le Conditions préalables pour démarrer pour configurer les autorisations requises et créer les données d'entraînement. Si vous avez également rempli leCommencer à utiliser un groupe de jeux de données de domaine (console), vous pouvez réutiliser les mêmes données sources. Si vous utilisez vos propres données sources, assurez-vous qu'elles sont formatées conformément aux prérequis.
- Configurez votre environnement SDK for Java 2.x AWS et vos informations d'identification comme indiqué dans la procédure de configuration AWS SDK for Java 2.x du manuel du développeur AWS SDK for Java 2.x .

didacticiel

Dans les étapes suivantes, vous allez configurer votre projet pour utiliser les packages Amazon Personalize et créer le SDK Amazon Personalize pour les clients Java 2.x. Ensuite, vous importez des données, vous créez un outil de recommandation pour le cas d'utilisation qui vous convient le mieux et vous obtenez des recommandations.

Étape 1 : configurer votre projet pour utiliser les packages Amazon Personalize

Une fois les conditions requises remplies, ajoutez les dépendances Amazon Personalize à votre fichier pom.xml et importez les packages Amazon Personalize.

1. Ajoutez les dépendances suivantes à votre fichier pom.xml. Les numéros de version les plus récents peuvent être différents de ceux de l'exemple de code.

```
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalize</artifactId>
 <version>2.16.83</version>
</dependency>
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalizeruntime</artifactId>
 <version>2.16.83</version>
</dependency>
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalizeevents</artifactId>
 <version>2.16.83</version>
</dependency>
```

2. Ajoutez les instructions d'importation suivantes à votre projet.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
 software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
 software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// recommender packages
import software.amazon.awssdk.services.personalize.model.CreateRecommenderRequest;
import software.amazon.awssdk.services.personalize.model.CreateRecommenderResponse;
import software.amazon.awssdk.services.personalize.model.DescribeRecommenderRequest;
// get recommendations packages
```

```
import
 software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
 software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

Étape 2 : créer des clients Amazon Personalize

Après avoir ajouté les dépendances Amazon Personalize à votre fichier pom.xml et importé les packages requis, créez les clients Amazon Personalize suivants :

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
  .region(region)
  .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
  .region(region)
  .build();
```

Étape 3 : Importer des données

Après avoir initialisé vos clients Amazon Personalize, importez les données historiques que vous avez créées lorsque vous avez terminé leConditions préalables pour démarrer. Pour importer des données historiques dans Amazon Personalize, procédez comme suit :

1. Enregistrez le schéma Avro suivant sous forme de fichier JSON dans votre répertoire de travail. Ce schéma correspond aux colonnes du fichier CSV que vous avez créé lorsque vous avez terminé leCréation des données d'entraînement (groupe de jeux de données de domaine).

   ```
   {
     "type": "record",
     "name": "Interactions",
     "namespace": "com.amazonaws.personalize.schema",
     "fields": [
         {
             "name": "USER_ID",
             "type": "string"
         },
         {
   ```

```
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "EVENT_TYPE",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ],
    "version": "1.0"
}
```

2. Utilisez la `createDomainSchema` méthode suivante pour créer un schéma de domaine dans Amazon Personalize. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de votre schéma, `VIDEO_ON_DEMAND` du domaine et le chemin du fichier JSON du schéma que vous avez créé à l'étape précédente. La méthode renvoie l'Amazon Resource Name (ARN) de votre nouveau schéma. Conservez-le en vue d'une utilisation ultérieure.

```
    public static String createDomainSchema(PersonalizeClient personalizeClient,
String schemaName, String domain,
            String filePath) {

        String schema = null;
        try {
            schema = new String(Files.readAllBytes(Paths.get(filePath)));
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }

        try {
            CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
                    .name(schemaName)
                    .domain(domain)
                    .schema(schema)
                    .build();

            String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

            System.out.println("Schema arn: " + schemaArn);
```

```
            return schemaArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

3. Créez un groupe d'ensemble de données. Utilisez la `createDomainDatasetGroup` méthode suivante pour créer un groupe de jeux de données de domaine. Transmettez les paramètres suivants : un client du service Amazon Personalize, un nom pour le groupe de jeux de données et un pass `VIDEO_ON_DEMAND` pour le domaine. La méthode renvoie l'ARN de votre nouveau groupe de données. Conservez-le en vue d'une utilisation ultérieure.

```
    public static String createDomainDatasetGroup(PersonalizeClient
personalizeClient,
            String datasetGroupName,
            String domain) {

        try {
            CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
                    .name(datasetGroupName)
                    .domain(domain)
                    .build();
            return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

4. Créez un jeu de données sur les interactions entre les articles. Utilisez la `createDataset` méthode suivante pour créer un jeu de données d'interactions entre articles. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de votre ensemble de données, l'ARN de votre schéma, l'ARN de votre groupe de jeux de données et le type `Interactions` de jeu de données. La méthode renvoie l'ARN de votre nouvel ensemble de données. Conservez-le en vue d'une utilisation ultérieure.

```
    public static String createDataset(PersonalizeClient personalizeClient,
            String datasetName,
            String datasetGroupArn,
            String datasetType,
            String schemaArn) {
        try {
            CreateDatasetRequest request = CreateDatasetRequest.builder()
                    .name(datasetName)
                    .datasetGroupArn(datasetGroupArn)
                    .datasetType(datasetType)
                    .schemaArn(schemaArn)
                    .build();

            String datasetArn = personalizeClient.createDataset(request)
                    .datasetArn();
            System.out.println("Dataset " + datasetName + " created.");
            return datasetArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

5. Importez vos données à l'aide d'une tâche d'importation de jeux de données. Utilisez la `createPersonalizeDatasetImportJob` méthode suivante pour créer une tâche d'importation de jeux de données.

   Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de la tâche et l'ARN de votre ensemble de données Interactions. Transmettez le chemin du compartiment Amazon S3 (`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`) dans lequel vous avez stocké les données d'entraînement, ainsi que l'ARN de votre rôle de service. Vous avez créé ce rôle dans le cadre duConditions préalables pour démarrer. La méthode renvoie l'ARN de votre tâche d'importation de jeu de données. Stockez-le éventuellement pour une utilisation ultérieure.

```
    public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
            String jobName,
            String datasetArn,
```

```
            String s3BucketPath,
            String roleArn) {

        long waitInMilliseconds = 60 * 1000;
        String status;
        String datasetImportJobArn;

        try {
            DataSource importDataSource = DataSource.builder()
                    .dataLocation(s3BucketPath)
                    .build();

            CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
                    .datasetArn(datasetArn)
                    .dataSource(importDataSource)
                    .jobName(jobName)
                    .roleArn(roleArn)
                    .build();

            datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
                    .datasetImportJobArn();
            DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
                    .datasetImportJobArn(datasetImportJobArn)
                    .build();

            long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                DatasetImportJob datasetImportJob = personalizeClient
                        .describeDatasetImportJob(describeDatasetImportJobRequest)
                        .datasetImportJob();

                status = datasetImportJob.status();
                System.out.println("Dataset import job status: " + status);

                if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
```

```
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return datasetImportJobArn;

        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

Étape 4 : Création d'un recommandeur

Une fois votre tâche d'importation de jeu de données terminée, vous êtes prêt à créer un recommandateur. Utilisez la `createRecommender` méthode suivante pour créer une recommandation. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom du recommandeur, le nom de ressource Amazon (ARN) de votre groupe de données et l'ARN de la `arn:aws:personalize:::recipe/aws-vod-top-picks` recette. La méthode renvoie l'ARN de votre nouveau recommandeur. Conservez-le en vue d'une utilisation ultérieure.

```
    public static String createRecommender(PersonalizeClient personalizeClient,
            String name,
            String datasetGroupArn,
            String recipeArn) {

        long maxTime = 0;
        long waitInMilliseconds = 30 * 1000; // 30 seconds
        String recommenderStatus = "";

        try {
            CreateRecommenderRequest createRecommenderRequest =
CreateRecommenderRequest.builder()
                    .datasetGroupArn(datasetGroupArn)
                    .name(name)
                    .recipeArn(recipeArn)
                    .build();

            CreateRecommenderResponse recommenderResponse = personalizeClient
                    .createRecommender(createRecommenderRequest);
            String recommenderArn = recommenderResponse.recommenderArn();
            System.out.println("The recommender ARN is " + recommenderArn);
```

```
            DescribeRecommenderRequest describeRecommenderRequest =
DescribeRecommenderRequest.builder()
                    .recommenderArn(recommenderArn)
                    .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                        .status();
                System.out.println("Recommender status: " + recommenderStatus);

                if (recommenderStatus.equals("ACTIVE") ||
recommenderStatus.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return recommenderArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

Étape 5 : Obtenir des recommandations

Une fois que vous avez créé un recommandeur, vous l'utilisez pour obtenir des recommandations. Utilisez la `getRecs` méthode suivante pour obtenir des recommandations pour un utilisateur. Transmettez comme paramètres un client d'exécution Amazon Personalize, le nom de ressource Amazon (ARN) du recommandeur que vous avez créé à l'étape précédente et un ID utilisateur (par exemple,123). La méthode imprime la liste des éléments recommandés à l'écran.

```
   public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
String recommenderArn,
           String userId) {

       try {
           GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
                   .recommenderArn(recommenderArn)
                   .numResults(20)
                   .userId(userId)
                   .build();

           GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
                   .getRecommendations(recommendationsRequest);
           List<PredictedItem> items = recommendationsResponse.itemList();

           for (PredictedItem item : items) {
               System.out.println("Item Id is : " + item.itemId());
               System.out.println("Item score is : " + item.score());
           }
       } catch (AwsServiceException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

Commencer à utiliser un groupe de jeux de données de domaine (SDK pour Python (Boto3))

Ce didacticiel explique comment utiliser le SDK pour Python (Boto3) afin de créer un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND. Dans ce didacticiel, vous allez créer un outil de recommandation pour le cas d'utilisation qui vous convient le mieux.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Conditions préalables
- didacticiel

- Commencer à utiliser les blocs-notes Amazon Personalize APIs avec Jupyter (iPython)

Conditions préalables

Les étapes suivantes sont prérequises pour l'utilisation des exemples Python de ce guide :

- Complétez le Conditions préalables pour démarrer pour configurer les autorisations requises et créer les données d'entraînement. Si vous utilisez vos propres données sources, assurez-vous qu'elles sont formatées conformément aux prérequis.
- Configurez votre AWS SDK pour Python (Boto3) environnement comme indiqué dansConfiguration du AWS Kits SDK.

didacticiel

Au cours des étapes suivantes, vous allez vérifier votre environnement et créer un SDK pour les clients Python (Boto3) pour Amazon Personalize. Ensuite, vous importez des données, vous créez un outil de recommandation pour le cas d'utilisation qui vous convient le mieux et vous obtenez des recommandations.

Étape 1 : Vérifiez votre environnement Python et créez des clients boto3

Une fois que vous avez effectué les prérequis, exécutez l'exemple Python suivant pour vérifier que votre environnement a été configuré correctement. Ce code crée également les clients Amazon Personalize boto3 que vous utilisez dans ce didacticiel. Si votre environnement est correctement configuré, une liste des recettes disponibles s'affiche et vous pouvez exécuter les autres exemples de ce didacticiel.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

Étape 2 : Importer des données

Après avoir créé les clients Amazon Personalize boto3 et vérifié votre environnement, importez les données historiques que vous avez créées lorsque vous avez terminé le. Conditions préalables pour démarrer Pour importer des données historiques dans Amazon Personalize, procédez comme suit :

1. Utilisez le code suivant pour créer un schéma dans Amazon Personalize. Remplacez `gs-domain-interactions-schema` par un nom pour le schéma.

```
import json
schema = {
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}

create_interactions_schema_response = personalize.create_schema(
    name='gs-domain-interactions-schema',
    schema=json.dumps(schema),
    domain='VIDEO_ON_DEMAND'
)

interactions_schema_arn = create_interactions_schema_response['schemaArn']
```

```
print(json.dumps(create_interactions_schema_response, indent=2))
```

2. Créez un fichier groupe d'ensembles de données avec le code suivant. Remplacez `dataset group name` par le nom du groupe de jeux de données.

```
response = personalize.create_dataset_group(
  name = 'dataset group name',
  domain = 'VIDEO_ON_DEMAND'
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

3. Créez un jeu de données d'interactions entre articles dans votre nouveau groupe de jeux de données à l'aide du code suivant. Donnez un nom à l'ensemble de données `schema_arn` et indiquez le et `dataset_group_arn` des étapes précédentes.

```
response = personalize.create_dataset(
    name = 'interactions-dataset-name',
    schemaArn = interactions_schema_arn,
    datasetGroupArn = dsg_arn,
    datasetType = 'INTERACTIONS'
)

dataset_arn = response['datasetArn']
```

4. Importez vos données à l'aide d'une tâche d'importation de jeux de données avec le code suivant. Le code utilise la méthode describe_dataset_import_job pour suivre l'état de la tâche.

   Transmettez les paramètres suivants : le nom de la tâche, celui `dataset_arn` de l'étape précédente, le chemin du compartiment Amazon S3 (`s3://bucket name/folder name/ratings.csv`) dans lequel vous avez stocké les données de formation et l'ARN de votre rôle de service IAM. Vous avez créé ce rôle dans le cadre duConditions préalables pour démarrer. Amazon Personalize a besoin d'une autorisation pour accéder au compartiment. Pour plus d'informations sur l'octroi d'accès, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.

```
import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation':'s3://amzn-s3-demo-bucket/filename.csv'},
    roleArn = 'role_arn'
)

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

Étape 4 : créer un recommandeur

Une fois votre tâche d'importation de jeu de données terminée, vous êtes prêt à créer un recommandateur. Utilisez le code suivant pour créer un recommendeur. Transmettez les paramètres suivants : le nom du recommandeur, le nom de ressource Amazon (ARN) de votre groupe de données et l'ARN `arn:aws:personalize:::recipe/aws-vod-top-picks` de la recette. Le code utilise la méthode describe_recommender pour suivre l'état du recommandeur.

```
import time
create_recommender_response = personalize.create_recommender(
  name = 'gs-python-top-picks',
```

```
  recipeArn = 'arn:aws:personalize:::recipe/aws-vod-top-picks',
  datasetGroupArn = dsg_arn
)
recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:

    version_response = personalize.describe_recommender(
        recommenderArn = recommender_arn
    )
    status = version_response["recommender"]["status"]

    if status == "ACTIVE":
        print("Creation succeeded for {}".format(recommender_arn))

    elif status == "CREATE FAILED":
        print("Creation failed for {}".format(recommender_arn))

    if status == "ACTIVE":
        break
    else:
        print("Recommender creation is still in progress")

    time.sleep(60)
```

Étape 5 : Obtenir des recommandations

Après avoir créé un recommandeur, vous pouvez l'utiliser pour obtenir des recommandations avec le code suivant. Transmettez comme paramètres le Amazon Resource Name (ARN) du recommandeur que vous avez créé à l'étape précédente, ainsi qu'un ID utilisateur (par exemple,123). La méthode imprime la liste des éléments recommandés.

```
response = personalizeRt.get_recommendations(
    recommenderArn = "arn:aws:personalize:us-west-2:014025156336:recommender/gs-python-
top-picks-89",
    userId = '123'
)
print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Commencer à utiliser les blocs-notes Amazon Personalize APIs avec Jupyter (iPython)

Pour commencer à créer des groupes de jeux de données de domaines avec les blocs-notes Jupyter, clonez ou téléchargez une série de blocs-notes situés dans le dossier notebooks_managed_domains du référentiel d'échantillons Amazon Personalize. Les carnets de notes vous expliquent comment importer des données de formation, créer un outil de recommandation et obtenir des recommandations avec Amazon Personalize.

> **Note**
>
> Avant de commencer avec les blocs-notes, assurez-vous de créer votre environnement en suivant les étapes du fichier README.md

Commencer à utiliser un groupe de jeux de données de domaine (SDK pour JavaScript v3)

Ce didacticiel explique comment utiliser la AWS SDK pour JavaScript version 3 pour créer un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND. Dans ce didacticiel, vous allez créer un outil de recommandation pour le cas d'utilisation qui vous convient le mieux.

Pour consulter le code utilisé dans ce didacticiel GitHub, consultez les exemples de code Amazon Personalize pour le SDK pour JavaScript v3 dans le référentiel d'exemples de code AWS SDK.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Prérequis
- didacticiel

Prérequis

Les étapes suivantes sont nécessaires pour suivre ce didacticiel :

- Complétez le Conditions préalables pour démarrer pour configurer les autorisations requises et créer les données d'entraînement. Si vous avez également rempli leCommencer à utiliser un

groupe de jeux de données de domaine (console), vous pouvez réutiliser les mêmes données sources. Si vous utilisez vos propres données sources, assurez-vous qu'elles sont formatées conformément aux prérequis.

- Configurez le SDK JavaScript et les AWS informations d'identification comme indiqué dans la JavaScript procédure de configuration du SDK du Guide du AWS SDK pour JavaScript développeur.

didacticiel

Dans les étapes suivantes, vous allez installer les dépendances requises. Ensuite, vous créez un groupe de jeux de données, vous importez des données, vous créez un outil de recommandation pour le cas d'utilisation qui vous convient le mieux et vous obtenez des recommandations.

Si vous utilisez Node.js, vous pouvez exécuter chaque exemple de code en l'enregistrant sous forme de JavaScript fichier, puis en l'exécutant`node <fileName.js>`.

Étape 1 : installer les dépendances Amazon Personalize

Une fois les conditions requises remplies, installez les dépendances Amazon Personalize suivantes :

- @aws -sdk/personnalisez le client
- @aws -sdk/ client-personalize-runtime
- @aws -sdk/ client-personalize-events (facultatif pour ce didacticiel, mais obligatoire si vous souhaitez enregistrer des événements après avoir créé votre recommandateur)

Voici un exemple de `package.json` fichier que vous pouvez utiliser. Pour installer les dépendances avec Node.js, accédez à l'endroit où vous avez enregistré le `package.json` fichier et exécutez-le`npm install`.

```
{
  "name": "personalize-js-project",
  "version": "1.0.0",
  "description": "personalize operations",
  "type": "module",
  "author": "Author Name <email@address.com>",
  "license": "ISC",
  "dependencies": {
    "@aws-sdk/client-personalize": "^3.350.0",
    "@aws-sdk/client-personalize-events": "^3.350.0",
```

```
    "@aws-sdk/client-personalize-runtime": "^3.350.0",
    "fs": "^0.0.1-security"
  },
  "compilerOptions": {
    "resolveJsonModule": true,
    "esModuleInterop": true
  }
}
```

Étape 2 : créer des clients Amazon Personalize

Après avoir installé les dépendances, créez vos clients Amazon Personalize. Dans ce didacticiel, les exemples de code supposent que vous créez les clients dans un fichier nommé `personalizeClients.js` stocké dans un répertoire nommé`libs`.

Voici un exemple de fichier `personalizeClient.js`.

```
import { PersonalizeClient } from "@aws-sdk/client-personalize";
import { PersonalizeRuntimeClient } from "@aws-sdk/client-personalize-runtime";
import { PersonalizeEventsClient } from "@aws-sdk/client-personalize-events";
// Set your AWS region.
const REGION = "region"; //e.g. "us-east-1"

const personalizeClient = new PersonalizeClient({ region: REGION});
const personalizeEventsClient = new PersonalizeEventsClient({ region: REGION});
const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: REGION});

export { personalizeClient, personalizeEventsClient, personalizeRuntimeClient };
```

Étape 3 : Importer des données

Après avoir créé vos clients Amazon Personalize, importez les données historiques que vous avez créées lorsque vous avez terminé leConditions préalables pour démarrer. Pour importer des données historiques dans Amazon Personalize, procédez comme suit :

1. Enregistrez le schéma Avro suivant sous forme de fichier JSON dans votre répertoire de travail. Ce schéma correspond aux colonnes du fichier CSV que vous avez créé lorsque vous avez terminé leCréation des données d'entraînement (groupe de jeux de données de domaine).

   ```
   {
     "type": "record",
     "name": "Interactions",
   ```

```
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

2. Créez un schéma de domaine dans Amazon Personalize avec le `createDomainSchema.js` code suivant. `SCHEMA_PATH`Remplacez-le par le chemin du fichier schema.json que vous venez de créer. Mettez `createSchemaParam` à jour le pour spécifier un nom pour le schéma et pour `domain` spécifier`VIDEO_ON_DEMAND`.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from "node:fs";

const schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = "TEST"; // for unit tests.
```

```
}

// Set the domain schema parameters.
export const createDomainSchemaParam = {
  name: "NAME" /* required */,
  schema: mySchema /* required */,
  domain:
    "DOMAIN" /* required for a domain dataset group, specify ECOMMERCE or
 VIDEO_ON_DEMAND */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSchemaCommand(createDomainSchemaParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

3. Créez un groupe de jeux de données de domaine dans Amazon Personalize avec le `createDomainDatasetGroup.js` code suivant. Mettez à jour le `domainDatasetGroupParams` pour spécifier un nom pour le groupe de jeux de données, et pour `domain` spécifier`VIDEO_ON_DEMAND`.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: "NAME" /* required */,
  domain:
    "DOMAIN" /* required for a domain dsg, specify ECOMMERCE or VIDEO_ON_DEMAND */,
};
```

```
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetGroupCommand(domainDatasetGroupParams),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

4. Créez un ensemble de données sur les interactions entre articles dans Amazon Personalize à l'aide `createDataset.js` du code suivant. Mettez à jour le `createDatasetParam` pour spécifier le nom de ressource Amazon (ARN) du groupe de jeux de données et du schéma que vous venez de créer, donnez un nom à l'ensemble de données et pour`datasetType`, spécifie`Interactions`.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  datasetType: "DATASET_TYPE" /* required */,
  name: "NAME" /* required */,
  schemaArn: "SCHEMA_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetCommand(createDatasetParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
```

```
  }
};
run();
```

5. Importez vos données avec le `createDatasetImportJob.js` code suivant. Mettez à jour le `datasetImportJobParam` pour spécifier les éléments suivants :
   - Spécifiez le nom de la tâche et spécifiez l'ARN de votre jeu de données Interactions.
   - Pour`dataLocation`, spécifiez le chemin du compartiment Amazon S3 (`s3://`*`https://amzn-s3-demo-bucket.s3.region-code.amazonaws.com`*`/`*`folder name`*`/ratings.csv`) dans lequel vous avez stocké les données d'entraînement.
   - Pour `roleArn` spécifier le nom de ressource Amazon pour votre rôle de service Amazon Personalize. Vous avez créé ce rôle dans le cadre duConditions préalables pour démarrer.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetImportJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: "DATASET_ARN" /* required */,
  dataSource: {
    /* required */
    dataLocation: "S3_PATH",
  },
  jobName: "NAME" /* required */,
  roleArn: "ROLE_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetImportJobCommand(datasetImportJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
```

```
};
run();
```

Étape 4 : créer un recommandeur

Une fois votre tâche d'importation de jeu de données terminée, vous êtes prêt à créer un recommandateur. Pour créer un recommandeur, utilisez le `createRecommender.js` code suivant. Mettez à jour le `createRecommenderParam` avec ce qui suit : Spécifiez un nom pour le recommandeur, spécifiez l'ARN de votre groupe de jeux de données et pour `recipeArn` specify`arn:aws:personalize:::recipe/aws-vod-top-picks`.

```
// Get service clients module and commands using ES6 syntax.
import { CreateRecommenderCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the recommender's parameters.
export const createRecommenderParam = {
  name: "NAME" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateRecommenderCommand(createRecommenderParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Étape 5 : Obtenir des recommandations

Après avoir créé un recommandeur, vous l'utilisez pour obtenir des recommandations. Utilisez le `getRecommendations.js` code suivant pour obtenir des recommandations pour un utilisateur. Mettez `getRecommendationsParam` à jour le pour spécifier l'ARN du recommandeur que vous avez créé à l'étape précédente et spécifiez un ID utilisateur (par exemple,`123`).

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  recommenderArn: "RECOMMENDER_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Commencer à utiliser un groupe de jeux de données personnalisé

> **Important**
>
> Dans ce didacticiel, vous allez créer une solution qui utilise l'entraînement automatique. Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, veillez à supprimer la solution lorsque vous avez terminé.

Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Ce guide de démarrage explique comment fournir des recommandations de films personnalisées à vos utilisateurs à l'aide d'un groupe de données personnalisé et de la Recette de personnalisation par l'utilisateur V2 recette. Le didacticiel utilise des données historiques comprenant 100 000 évaluations de films sur 9 700 films de 600 utilisateurs.

Pour commencer, complétez le, Conditions préalables pour démarrer puis passez à Démarrer (console)Démarrer (AWS CLI),Mise en route (SDK pour Python (Boto3)), ouMise en route (SDK pour Java 2.x).

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Démarrer (console)
- Démarrer (AWS CLI)
- Mise en route (SDK pour Python (Boto3))
- Mise en route (SDK pour Java 2.x)

Démarrer (console)

Dans cet exercice, vous allez utiliser la console Amazon Personalize pour créer un groupe de données personnalisé avec une solution qui renvoie des recommandations de films pour un utilisateur donné. Avant de commencer cet exercice, passez en revue leConditions préalables pour démarrer.

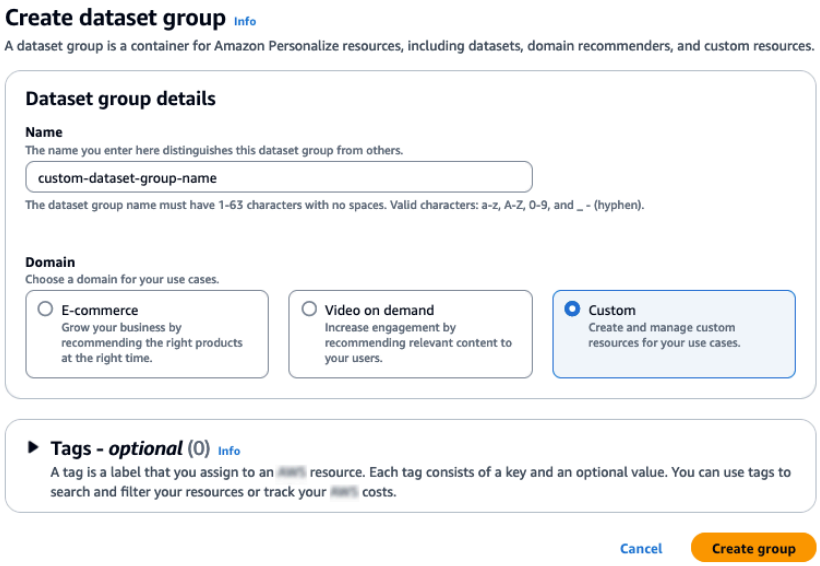
Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour plus d’informations, consultez Conditions requises pour supprimer des ressources Amazon Personalize.

Étape 1 : Création d'un groupe de jeux de données et d'un ensemble de données

Dans cette procédure, vous devez d'abord créer un groupe d’ensemble de données. Ensuite, vous créez un ensemble de données sur les interactions avec Amazon Personalize Item dans le groupe de jeux de données.

Pour créer un groupe de jeux de données et un ensemble de données

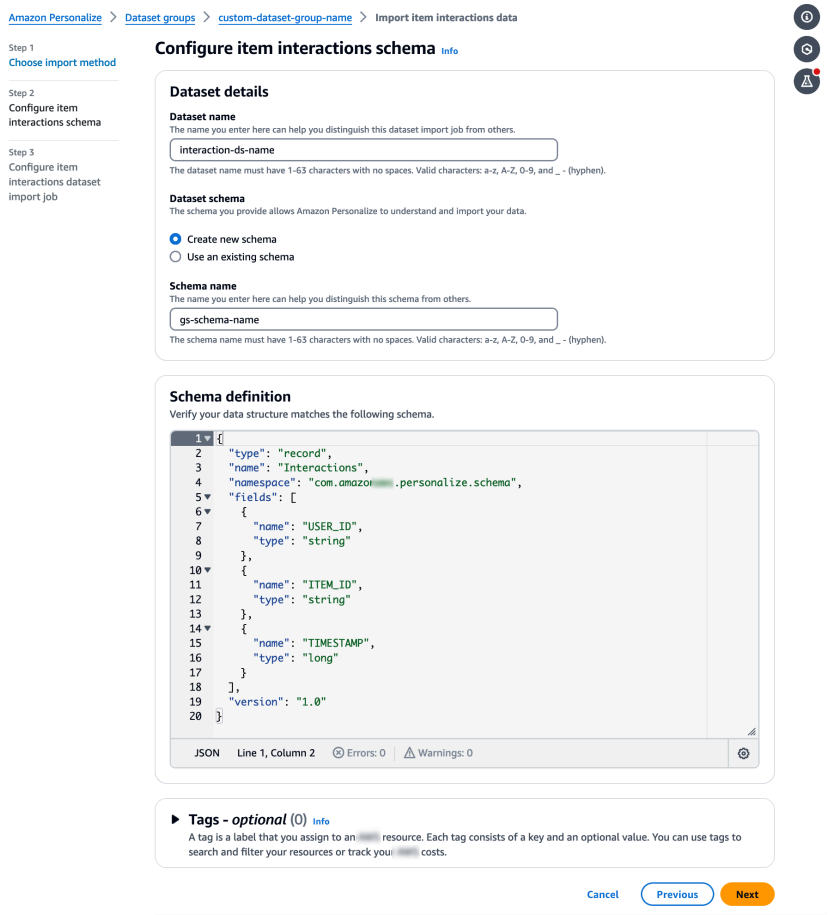
1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez Create dataset group (Créer un groupe d'ensembles de données).
3. Sous Dataset group details (Informations du groupe d'ensembles de données), pour Dataset group name (Nom du groupe d'ensembles de données), spécifiez un nom pour votre groupe d'ensembles de données.
4. Pour le domaine, choisissez Personnalisé. Votre écran doit se présenter comme suit :



5. Choisissez Créer un groupe. La page d'aperçu apparaît.
6. À l'étape 1. Créez des ensembles de données et importez des données, choisissez Create dataset et choisissez Item interactions dataset.
7. Choisissez Importer les données directement dans les ensembles de données Amazon Personalize, puis Next.
8. Sur la page Configurer le jeu de données des interactions entre les éléments, dans Nom du jeu de données, spécifiez un nom pour votre ensemble de données.
9. Pour le schéma du jeu de données, choisissez Créer un nouveau schéma. Dans la section Définition du schéma, un schéma minimal d'interactions entre les éléments est affiché. Le

schéma correspond aux en-têtes que vous avez précédemment ajoutés au `ratings.csv` fichier. Vous n'avez donc pas besoin d'apporter de modifications. Si vous n'avez pas créé les données d'entraînement, consultezConditions préalables pour démarrer.

10. Dans Nom du schéma, spécifiez le nom du nouveau schéma. Votre écran doit se présenter comme suit :



11. Choisissez Suivant. La page de tâche d'importation du jeu de données Configurer les interactions entre les articles apparaît. Ensuite, complétez Étape 2 : Importer les données relatives aux interactions entre les articles pour importer les données d'interactions.

Étape 2 : Importer les données relatives aux interactions entre les articles

Maintenant que vous avez créé un jeu de données, il est temps d'importer les données relatives aux interactions entre les éléments dans le jeu de données.

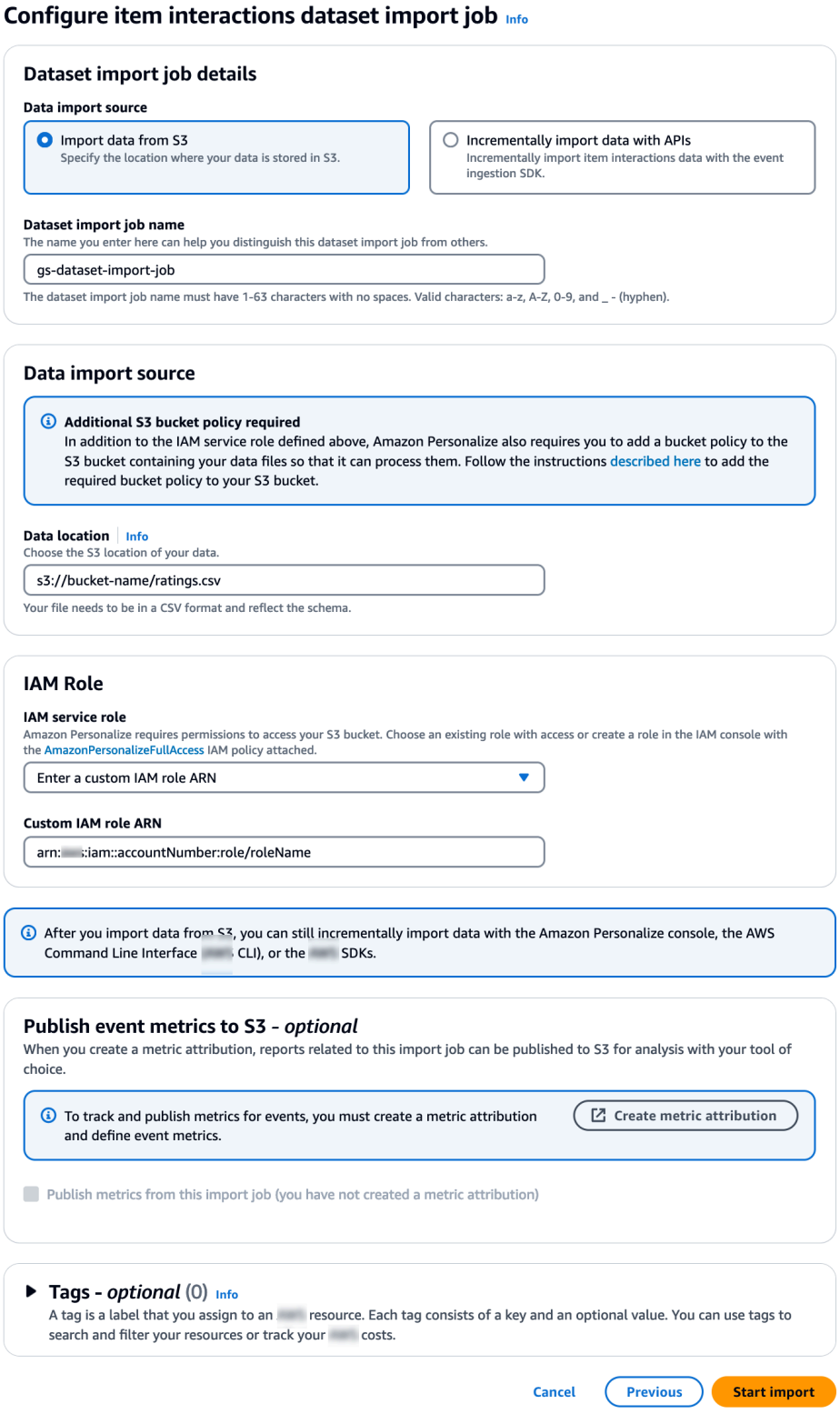
Pour importer des données sur les interactions entre articles

1. Sur la page de la tâche d'importation du jeu de données Configurer les interactions entre les éléments, pour Source d'importation de données, sélectionnez Importer les données depuis S3.
2. Pour le nom de la tâche d'importation du jeu de données, spécifiez un nom pour votre tâche d'importation.
3. Dans la boîte de dialogue Politique de compartiment S3 supplémentaire requise, si vous n'avez pas accordé les autorisations Amazon Personalize, suivez les instructions pour ajouter la politique de compartiment Amazon S3 requise.
4. Pour Emplacement des données, spécifiez où le fichier de données de votre film est stocké dans Amazon Simple Storage Service (S3). Utilisez la syntaxe suivante :

   ```
   s3://amzn-s3-demo-bucket/<folder path>/filename.csv
   ```

5. Dans la section Rôle IAM, pour le rôle de service IAM, choisissez Enter a custom IAM role ARN.
6. Pour Custom IAM role ARN (ARN du rôle IAM personnalisé), spécifiez le rôle que vous avez créé dans Création d'un rôle IAM pour Amazon Personalize.

   Les sections relatives aux détails de la tâche d'importation du jeu de données et au rôle IAM doivent être similaires aux suivantes :



7. Laissez les sections Publier les métriques des événements dans S3 et Tags inchangées et choisissez Démarrer l'importation. La tâche d'importation de données démarre et la page

d'aperçu s'affiche. Au départ, le statut est Création en attente (suivi de Création en cours) et le bouton Créer une solution est désactivé.

Lorsque le travail d'importation de données est terminé, le statut passe à Actif et le bouton Créer une solution est activé.

8. Maintenant que vous avez importé des données, vous êtes prêt à créer une solution dansÉtape 3 : Création d'une solution.

Étape 3 : Création d'une solution

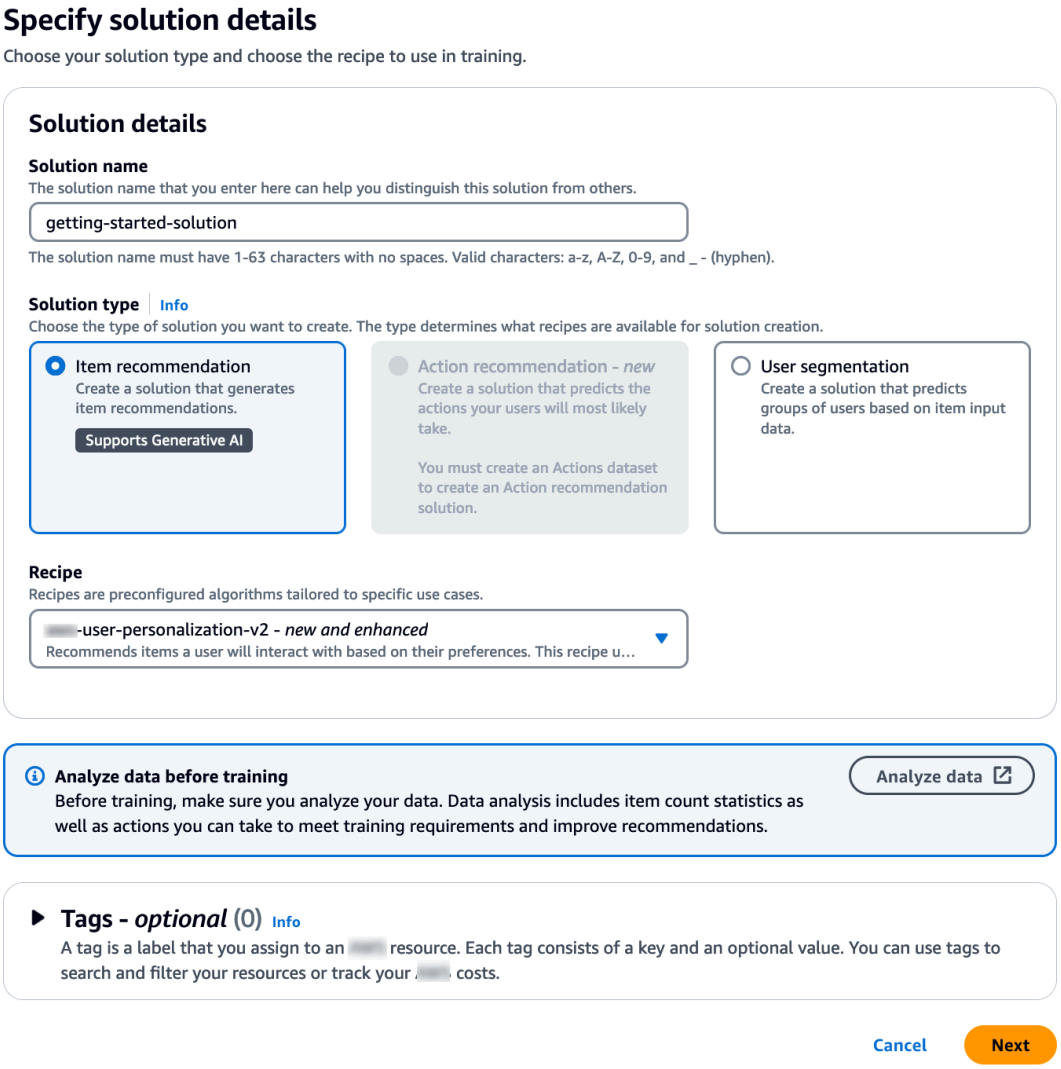
Dans ce didacticiel, vous allez utiliser le jeu de données dans lequel vous avez importé Étape 2 : Importer les données relatives aux interactions entre les articles pour entraîner un modèle. Un modèle formé est appelé version de solutions.

> **Important**
>
> Dans ce didacticiel, vous allez créer une solution qui utilise l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, veillez à supprimer la solution lorsque vous avez terminé. Pour plus d'informations, consultez Conditions requises pour supprimer des ressources Amazon Personalize.

Pour créer une solution

1. Sur la page Vue d'ensemble de votre groupe de jeux de données, à l'étape 3. Configurez des ressources de formation et de recommandation, choisissez Create solutions.
2. Pour Solution name (Nom de la solution), spécifiez un nom pour votre solution.
3. Dans Type de solution, sélectionnez Recommandations relatives aux articles.
4. Pour Recipe, choisissez`aws-user-personalization-v2`.

   Votre écran doit se présenter comme suit :



5. Choisissez Suivant. Laissez les champs de configuration de formation inchangés. La solution que vous créez entraîne automatiquement les nouveaux modèles tous les 7 jours et accorde plus de poids aux données les plus récentes sur les interactions entre les articles.
6. Choisissez Next et passez en revue les détails de la solution.
7. Choisissez Créer une solution et la page de détails de la solution s'affiche. Une fois que vous avez créé une solution, Amazon Personalize commence à créer votre première version de solution en moins d'une heure. Lorsque la formation commence, elle apparaît dans la section Versions de la solution sur la page de détails et vous pouvez surveiller son état.

   Lorsque le statut de la version de la solution est Actif, vous êtes prêt à passer àÉtape 4 : Créer une campagne.

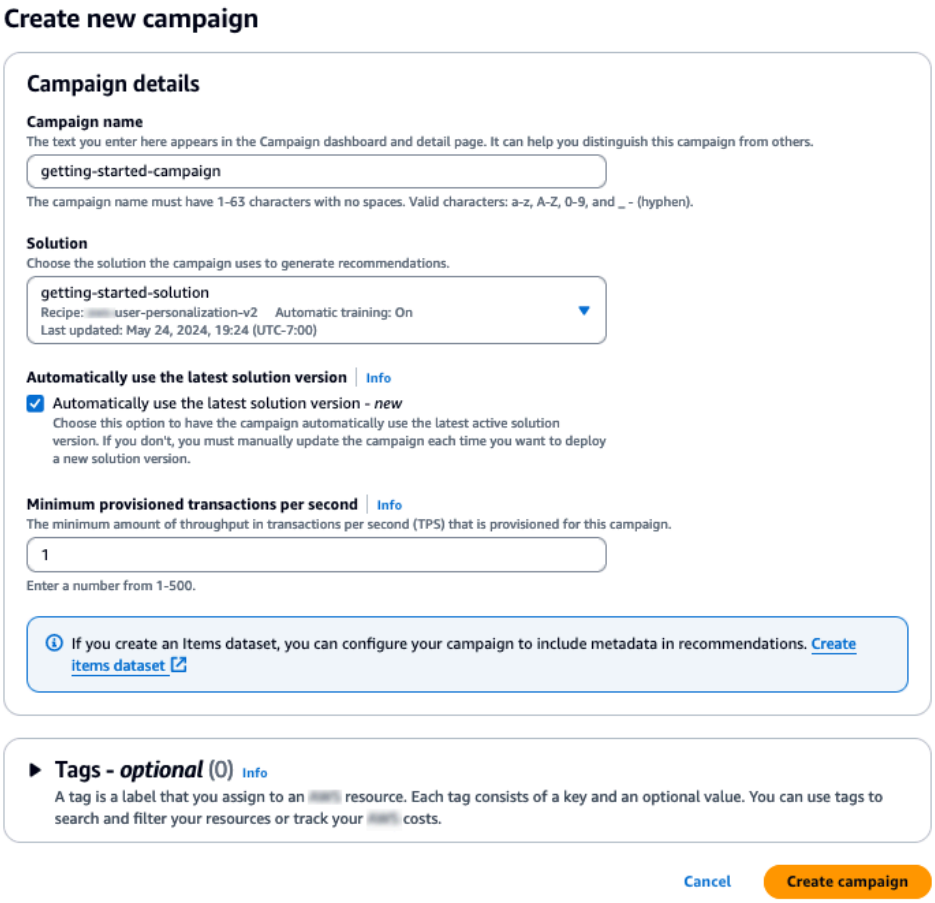
Étape 4 : Créer une campagne

Dans cette procédure, vous créez une campagne qui déploie la version de solution que vous avez créée à l'étape précédente.

Création d'une campagne

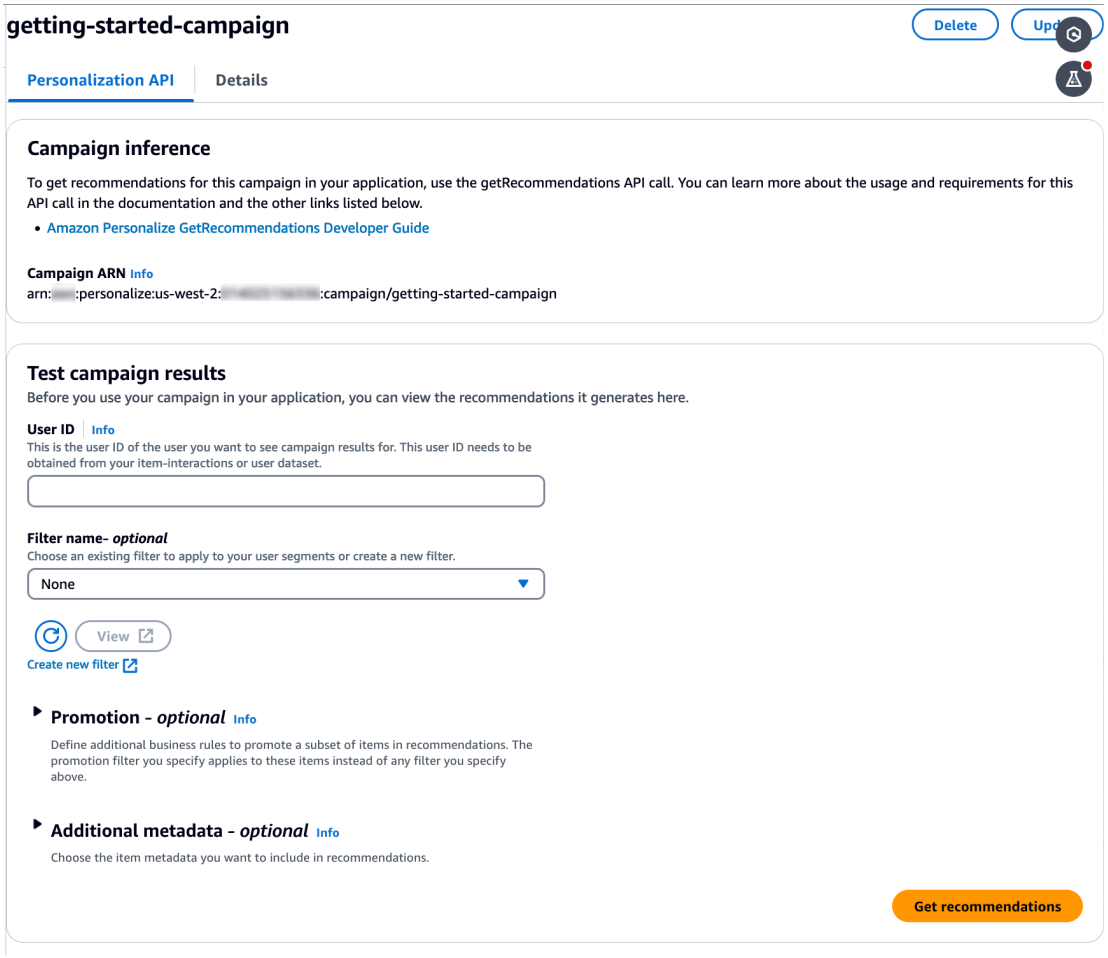
1. Dans le volet de navigation, développez les ressources personnalisées et choisissez Campaigns.
2. Choisissez Create campaign (Créer une campagne). La page Créer une nouvelle campagne apparaît.
3. Dans Campaign details (Détails de la campagne), pour Campaign name (nom de la campagne), spécifiez un nom pour votre campagne.
4. Dans Solution, choisissez la solution que vous avez créée à l'étape précédente.
5. Choisissez Utiliser automatiquement la dernière version de la solution. Laissez tous les autres champs inchangés.

   Votre écran doit se présenter comme suit :



6. Choisissez Create campaign (Créer une campagne). La création de la campagne commence et les pages de détails de la campagne contenant la section API de personnalisation s'affichent.

   La création d'une campagne peut prendre quelques minutes. Une fois qu'Amazon Personalize a fini de créer votre campagne, la page est mise à jour pour afficher la section des résultats des campagnes de test. Votre écran doit se présenter comme suit :



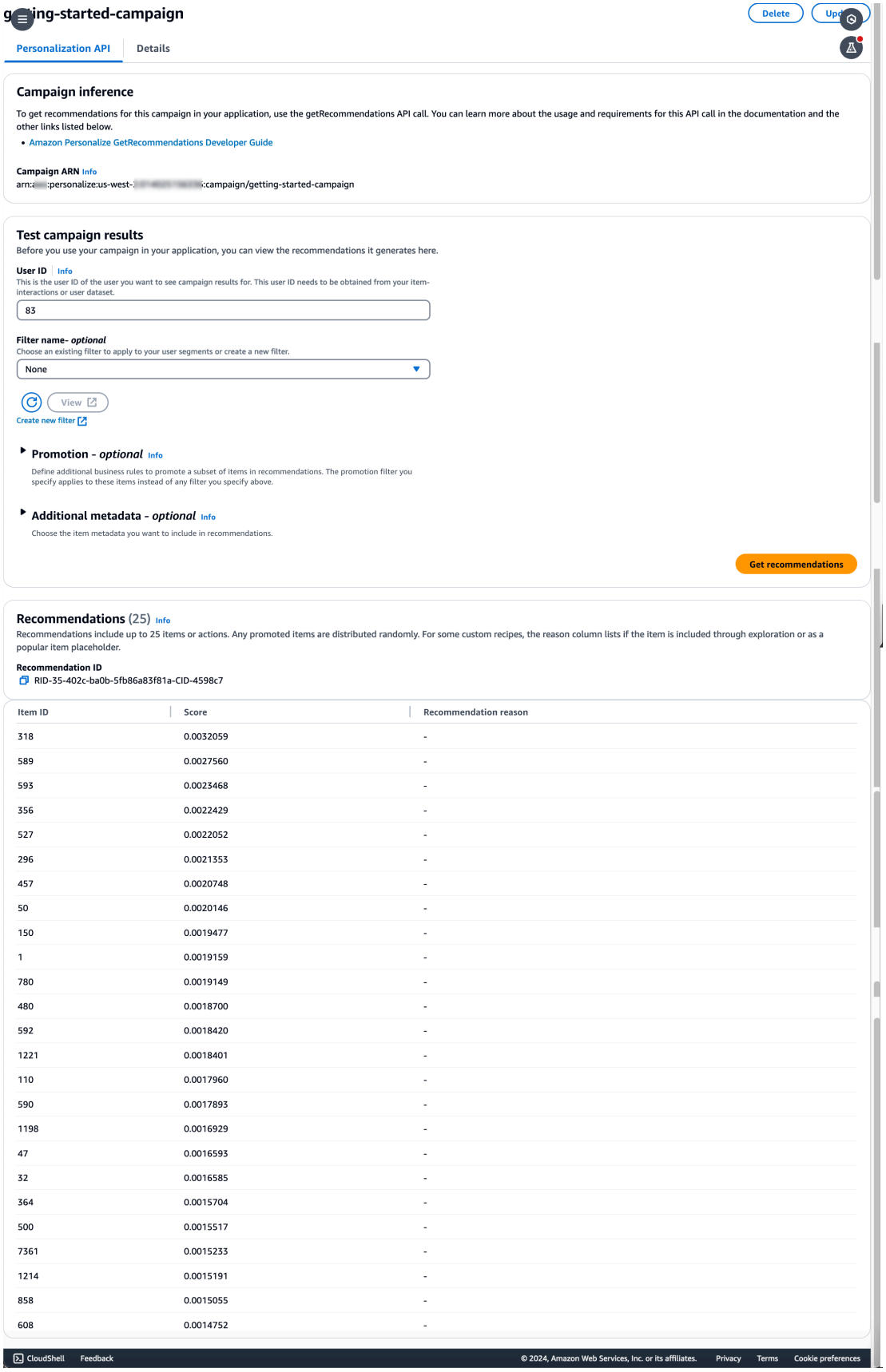
Étape 5 : Obtenir des recommandations

Dans cette procédure, vous devez utiliser la campagne que vous avez créée à l'étape précédente pour obtenir des recommandations.

Obtention de recommandations

1. Sous Test campaign results (Résultats de la campagne de test), pour User ID (ID utilisateur), spécifiez une valeur de l'ensemble de données ratings (évaluations), par exemple, **83**. Laissez tous les autres champs inchangés.
2. Choisissez Obtenir des recommandations. Le panneau Recommandations répertorie les identifiants et les scores des éléments recommandés.

   Votre écran doit se présenter comme suit :


ing-started-campaign
Delete
Personalization API
Details
Campaign inference
To get recommendations for this campaign in your application, use the getRecommendations API call. You can learn more about the usage and requirements for this API call in the documentation and the other links listed below.
Amazon Personalize GetRecommendations Developer Guide
Campaign ARN Info
:personalize:us-west-
:campaign/getting-started-campaign
Test campaign results
Before you use your campaign in your application, you can view the recommendations it generates here.
User ID Info
This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your item-interactions or user dataset.
83
Filter name- optional
Choose an existing filter to apply to your user segments or create a new filter.
None
View
Create new filter
Promotion - optional Info
Define additional business rules to promote a subset of items in recommendations. The promotion filter you specify applies to these items instead of any filter you specify above.
Additional metadata - optional Info
Choose the item metadata you want to include in recommendations.
Get recommendations
Recommendations (25) Info
Recommendations include up to 25 items or actions. Any promoted items are distributed randomly. For some custom recipes, the reason column lists if the item is included through exploration or as a popular item placeholder.
Recommendation ID
RID-35-402c-ba0b-5fb86a83f81a-CID-4598c7
Item ID | Score | Recommendation reason
318 | 0.0032059 | -
589 | 0.0027560 | -
593 | 0.0023468 | -
356 | 0.0022429 | -
527 | 0.0022052 | -
296 | 0.0021353 | -
457 | 0.0020748 | -
50 | 0.0020146 | -
150 | 0.0019477 | -
1 | 0.0019159 | -
780 | 0.0019149 | -
480 | 0.0018700 | -
592 | 0.0018420 | -
1221 | 0.0018401 | -
110 | 0.0017960 | -
590 | 0.0017893 | -
1198 | 0.0016929 | -
47 | 0.0016593 | -
32 | 0.0016585 | -
364 | 0.0015704 | -
500 | 0.0015517 | -
7361 | 0.0015233 | -
1214 | 0.0015191 | -
858 | 0.0015055 | -
608 | 0.0014752 | -
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Démarrer (AWS CLI)

Dans cet exercice, vous allez utiliser le AWS Command Line Interface (AWS CLI) pour découvrir Amazon Personalize. Vous créerez une campagne qui retourne des recommandations de films pour un ID utilisateur donné.

Avant de commencer cet exercice, procédez comme suit :

- Consultez les Conditions préalables pour démarrer de démarrage.
- Configurez le AWS CLI, comme indiqué dansConfiguration du AWS CLI.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

> **Note**
>
> Les AWS CLI commandes de cet exercice ont été testées sous Linux. Pour plus d'informations sur l'utilisation des AWS CLI commandes sous Windows, reportez-vous à la section Spécification des valeurs de paramètres pour le AWS Command Line Interface dans le Guide de AWS Command Line Interface l'utilisateur.

Étape 1 : Importer les données d'entraînement

Suivez ces étapes pour créer un groupe d'ensembles de données, ajouter un ensemble de données au groupe, puis remplir l'ensemble de données à l'aide des données de notation des films.

1. Créez un groupe d'ensembles de données en exécutant la commande suivante. Vous pouvez chiffrer le groupe de jeux de données en transmettant un ARN AWS Key Management Serviceclé et l'ARN d'un rôle IAM autorisé à accéder à cette clé en tant que paramètres d'entrée. Pour plus d'informations sur l'API, consultez CreateDatasetGroup.

   ```
   aws personalize create-dataset-group --name MovieRatingDatasetGroup --kms-key-
   arn arn:aws:kms:us-west-2:01234567890:key/1682a1e7-a94d-4d92-bbdf-837d3b62315e --
   role-arn arn:aws:iam::01234567890:KMS-key-access
   ```

   L'ARN du groupe d'ensembles de données est affiché, par exemple :

```
{
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup"
}
```

Utilisez la commande `describe-dataset-group` pour afficher le groupe d'ensembles de données que vous avez créé en spécifiant l'ARN du groupe d'ensembles de données renvoyé.

```
aws personalize describe-dataset-group \
--dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup
```

Le groupe d'ensembles de données et ses propriétés s'affichent, par exemple :

```
{
    "datasetGroup": {
        "name": "MovieRatingDatasetGroup",
        "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
        "status": "ACTIVE",
        "creationDateTime": 1542392161.262,
        "lastUpdatedDateTime": 1542396513.377
    }
}
```

> **Note**
>
> Attendez que le `status` du groupe d'ensembles de données apparaisse comme ACTIVE (ACTIF) avant de créer un ensemble de données dans le groupe. Cette opération est généralement rapide.

Si vous ne vous souvenez pas de l'ARN du groupe de jeux de données, utilisez la `list-dataset-groups` commande pour afficher tous les groupes de jeux de données que vous avez créés, ainsi que leurs ARNs.

```
aws personalize list-dataset-groups
```

> **Note**
>
> Les `list-objects` commandes `describe-object` et sont disponibles pour la plupart des objets Amazon Personalize. Ces commandes n'apparaissent pas dans le reste de cet exercice, mais elles sont disponibles.

2. Créez un fichier de schéma au format JSON en enregistrant le code suivant dans un fichier nommé `MovieRatingSchema.json`. Le schéma correspond aux en-têtes que vous avez précédemment ajoutés à `ratings.csv`. Le nom du schéma est`Interactions`, qui correspond à l'un des types de jeux de données reconnus par Amazon Personalize. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

3. Créez un schéma en exécutant la commande suivante. Spécifiez le fichier que vous avez enregistré lors de l'étape précédente. L'exemple montre que le fichier appartient au dossier actif. Pour plus d’informations sur l’API, consultez CreateSchema.

```
aws personalize create-schema \
  --name MovieRatingSchema \
```

```
  --schema file://MovieRatingSchema.json
```

Le nom Amazon Resource Name (ARN) du schéma est affiché, par exemple :

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema"
}
```

4. Créez un ensemble de données vide en exécutant la commande suivante. Fournissez l'ARN du groupe d'ensembles de données et l'ARN du schéma qui ont été retournés lors des étapes précédentes. Le `dataset-type` doit correspondre au schéma `name` de l'étape précédente. Pour plus d'informations sur l'API, consultez CreateDataset.

```
aws personalize create-dataset \
  --name MovieRatingDataset \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --dataset-type Interactions \
  --schema-arn arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema
```

L'ARN de l'ensemble de données est affiché, par exemple :

```
{
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS"
}
```

5. Ajoutez les données de formation à l'ensemble de données :

   a. Créez une tâche d'importation d'ensembles de données en exécutant la commande suivante. Indiquez l'ARN du jeu de données et le nom du compartiment Amazon S3 renvoyés lors des étapes précédentes. Indiquez l'ARN du rôle Gestion des identités et des accès AWS (IAM) dans lequel vous l'avez créé. Création d'un rôle IAM pour Amazon Personalize Pour plus d'informations sur l'API, consultez CreateDatasetImportJob.

```
aws personalize create-dataset-import-job \
  --job-name MovieRatingImportJob \
  --dataset-arn arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS \
  --data-source dataLocation=s3://amzn-s3-demo-bucket/ratings.csv \
```

```
  --role-arn roleArn
```

L'ARN de la tâche d'importation d'ensembles de données est affiché, par exemple :

```
{
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-
job/MovieRatingImportJob"
}
```

b. Vérifiez le statut à l'aide de la commande `describe-dataset-import-job`. Fournissez l'ARN de la tâche d'importation d'ensembles de données qui a été retourné lors de l'étape précédente. Pour plus d'informations sur l'API, consultez DescribeDatasetImportJob.

```
aws personalize describe-dataset-import-job \
  --dataset-import-job-arn arn:aws:personalize:us-west-2:acct-id:dataset-
import-job/MovieRatingImportJob
```

Les propriétés de la tâche d'importation d'ensembles de données et son statut sont affichés. Initialement, le `status` indique CREATE PENDING (CRÉATION EN ATTENTE), par exemple :

```
{
  "datasetImportJob": {
      "jobName": "MovieRatingImportJob",
      "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-
import-job/MovieRatingImportJob",
      "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS",
      "dataSource": {
          "dataLocation": "s3://amzn-s3-demo-bucket/ratings.csv"
      },
      "roleArn": "role-arn",
      "status": "CREATE PENDING",
      "creationDateTime": 1542392161.837,
      "lastUpdatedDateTime": 1542393013.377
  }
}
```

L'importation d'ensembles de données est terminée quand le statut ACTIVE (ACTIF) apparaît. Vous êtes alors prêt à former le modèle à l'aide de l'ensemble de données spécifié.

> **Note**
>
> L'importation nécessite un certain temps. Attendez que l'importation d'ensembles de données soit terminée avant de former le modèle à l'aide de l'ensemble de données.

Étape 2 : créer une solution (entraîner le modèle)

Pour entraîner un modèle, vous créez la configuration pour entraîner le modèle à l'aide de l'CreateSolutionopération et vous laissez l'entraînement automatique activé. La solution commence automatiquement à entraîner la première solution en moins d'une heure.

Vous entraînez un modèle à l'aide d'une recette et de vos données d'entraînement. Amazon Personalize propose un ensemble de recettes prédéfinies. Pour de plus amples informations, veuillez consulter Choisir une recette. Pour cet exercice, vous utilisez la recette User-Personalization-v 2.

1. Créez la configuration pour la formation d'un modèle en exécutant la commande suivante. Cette commande crée une solution qui utilise l'entraînement automatique. Il crée automatiquement une nouvelle version de solution tous les sept jours (par défaut).

```
aws personalize create-solution \
  --name MovieSolution \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --recipe-arn arn:aws:personalize:::recipe/aws-user-personalization-v2 \
  --perform-auto-training \
  --solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(7
 days)\"}}"
```

   L'ARN de la solution est affiché, par exemple :

```
{
  "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution"
}
```

2. Vérifiez le statut de création à l'aide de la commande `describe-solution`. Fournissez l'ARN de la solution qui a été retourné lors de l'étape précédente. Pour plus d'informations sur l'API, consultez DescribeSolution.

```
aws personalize describe-solution \
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

Les propriétés de la solution et du `status` de création sont affichées. Par exemple :

```
{
  "solution": {
      "name": "MovieSolution",
      "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution",
      "performHPO": false,
      "performAutoML": false,
      "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization-v2",
      "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
      "solutionConfig": {
            "algorithmHyperParameters": {
                "apply_recency_bias": "true"
            },
            "featureTransformationParameters": {},
            "autoTrainingConfig": {
                "schedulingExpression": "rate(7 days)"
            }
       },
      "status": "ACTIVE",
      "creationDateTime": "2021-05-12T16:27:59.819000-07:00",
      "lastUpdatedDateTime": "2021-05-12T16:27:59.819000-07:00"
  }
}
```

3. Avec l'entraînement automatique, l'entraînement à la version de la solution commence une fois que la solution est ACTIVE. Après le début de la formation, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution à l'aide de la ListSolutionVersionscommande suivante :

   ```
   aws personalize list-solution-versions --solution-arn arn:aws:personalize:us-
   west-2:acct-id:solution/MovieSolution
   ```

4. Vérifiez le statut de formation de la version de solution à l'aide de la commande `describe-solution-version`. Fournissez l'ARN de la version de solution qui a été renvoyé lors de l'étape précédente. Pour plus d'informations sur l'API, consultez DescribeSolutionVersion.

```
aws personalize describe-solution-version \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/version-id
```

Les propriétés de la version de solution et le `status` de formation sont affichés. Initialement, le statut indique CREATE PENDING (CRÉATION EN ATTENTE), par exemple :

```
{
  "solutionVersion": {
      "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
      ...,
      "status": "CREATE PENDING"
  }
}
```

5. Lorsque la version de la solution `status` est ACTIVE, la formation est terminée.

   Vous pouvez désormais consulter les indicateurs de formation et créer une campagne à l'aide de la version de la solution.

   > **Note**
   >
   > La formation nécessite un certain temps. Attendez que la formation soit terminée (le statut de formation indique ACTIVE (ACTIF)) avant d'utiliser cette version de la solution dans une campagne.

6. Vous pouvez valider les performances de la version de solutions en passant en revue ses métriques. Obtenez les métriques de la version de solutions en exécutant la commande suivante. Fournissez l'ARN de la version de solution qui a été renvoyée précédemment. Pour plus d'informations sur l'API, consultez GetSolutionMetrics.

   ```
   aws personalize get-solution-metrics \
     --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
   MovieSolution/version-id
   ```

   Voici un exemple de réponse :

   ```
   {
   ```

```
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/www-
solution/<version-id>",
  "metrics": {
        "coverage": 0.0485,
        "mean_reciprocal_rank_at_25": 0.0381,
        "normalized_discounted_cumulative_gain_at_10": 0.0363,
        "normalized_discounted_cumulative_gain_at_25": 0.0984,
        "normalized_discounted_cumulative_gain_at_5": 0.0175,
        "precision_at_10": 0.0107,
        "precision_at_25": 0.0207,
        "precision_at_5": 0.0107
    }
}
```

Étape 3 : Création d'une campagne (déploiement de la solution)

Avant de pouvoir obtenir des recommandations, vous devez déployer une version de solution. Le déploiement d'une solution est également appelé la création d'une campagne. Une fois que vous avez créé votre campagne, votre application client peut obtenir des recommandations à l'aide l'API GetRecommendations.

1. Créez une campagne en exécutant la commande suivante. Fournissez l'ARN de la version de solution qui a été renvoyé lors de l'étape précédente. Pour plus d'informations sur l'API, consultez CreateCampaign.

   ```
   aws personalize create-campaign \
     --name MovieRecommendationCampaign \
     --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
   MovieSolution/version-id \
     --min-provisioned-tps 1
   ```

   Voici un exemple de réponse :

   ```
   {
     "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
   MovieRecommendationCampaign"
   }
   ```

2. Vérifiez le statut de déploiement en exécutant la commande suivante. Fournissez l'ARN de la campagne qui a été retourné lors de l'étape précédente. Pour plus d'informations sur l'API, consultez DescribeCampaign.

```
aws personalize describe-campaign \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign
```

Voici un exemple de réponse :

```
{
  "campaign": {
      "name": "MovieRecommendationCampaign",
      "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign",
      "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
      "minProvisionedTPS": "1",
      "creationDateTime": 1543864775.923,
      "lastUpdatedDateTime": 1543864791.923,
      "status": "CREATE IN_PROGRESS"
  }
}
```

> **Note**
>
> Attendez que le `status` apparaisse comme ACTIVE (ACTIF) avant d'obtenir des recommandations à partir de la campagne.

Étape 4 : Obtenir des recommandations

Obtenez des recommandations en exécutant la commande `get-recommendations`. Fournissez l'ARN de la campagne qui a été retourné lors de l'étape précédente. Dans la requête, spécifiez un ID utilisateur à partir de l'ensemble de données de notation des films. Pour plus d'informations sur l'API, consultez GetRecommendations.

> **Note**
>
> Toutes les recettes ne prennent pas en charge l'API `GetRecommendations`. Pour de plus amples informations, veuillez consulter Choisir une recette.
> La AWS CLI commande que vous appelez à cette étape est différente de celle des étapes précédentes. `personalize-runtime`

```
aws personalize-runtime get-recommendations \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign \
  --user-id 123
```

En réponse, la campagne renvoie une liste de recommandations d'articles (films IDs) susceptibles d'intéresser l'utilisateur. La liste est triée par ordre décroissant de pertinence pour l'utilisateur.

```
{
  "itemList": [
      {
          "itemId": "14"
      },
      {
          "itemId": "15"
      },
      {
          "itemId": "275"
      },
      {
          "itemId": "283"
      },
      {
          "itemId": "273"
      },
      ...
  ]
}
```

Mise en route (SDK pour Python (Boto3))

Ce didacticiel explique comment effectuer le flux de travail Amazon Personalize du début à la fin avec le SDK pour Python (Boto3).

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Rubriques

- Conditions préalables
- didacticiel
- Commencer à utiliser les blocs-notes Amazon Personalize APIs avec Jupyter (iPython)

Conditions préalables

Les étapes suivantes sont prérequises pour l'utilisation des exemples Python de ce guide :

- Complétez le Conditions préalables pour démarrer pour configurer les autorisations requises et créer les données d'entraînement. Si vous utilisez vos propres données sources, assurez-vous qu'elles sont formatées conformément aux prérequis.
- Configurez votre AWS SDK pour Python (Boto3) environnement comme indiqué dansConfiguration du AWS Kits SDK.

didacticiel

Au cours des étapes suivantes, vous allez vérifier votre environnement et créer un SDK pour les clients Python (Boto3) pour Amazon Personalize. Ensuite, vous importez des données, vous créez et déployez une version de solution avec une campagne et vous obtenez des recommandations.

Étape 1 : Vérifiez votre environnement Python et créez des clients boto3

Une fois que vous avez effectué les prérequis, exécutez l'exemple Python suivant pour vérifier que votre environnement a été configuré correctement. Ce code crée également les clients Amazon Personalize boto3 que vous utilisez dans ce didacticiel. Si votre environnement est correctement configuré, la liste des recettes disponibles s'affiche et vous pouvez exécuter les autres exemples de ce didacticiel.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

Étape 2 : Importer des données

Après avoir créé les clients Amazon Personalize boto3 et vérifié votre environnement, importez les données historiques que vous avez créées lorsque vous avez terminé le. Conditions préalables pour démarrer Pour importer des données historiques dans Amazon Personalize, procédez comme suit :

1. Utilisez le code suivant pour créer un schéma dans Amazon Personalize. Remplacez `getting-started-schema` par un nom pour le schéma.

```
import json
schema = {
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}

create_interactions_schema_response = personalize.create_schema(
```

```
    name='getting-started-schema',
    schema=json.dumps(schema)
)

interactions_schema_arn = create_interactions_schema_response['schemaArn']
print(json.dumps(create_interactions_schema_response, indent=2))
```

2. Créez un fichier groupe d'ensembles de données avec le code suivant. Remplacez `dataset group name` par un nom pour le groupe de jeux de données.

```
response = personalize.create_dataset_group(name = 'dataset group name')
dataset_group_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dataset_group_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

3. Créez un jeu de données d'interactions entre articles dans votre nouveau groupe de jeux de données à l'aide du code suivant. Donnez un nom à l'ensemble de données `schema_arn` et indiquez le et `dataset_group_arn` des étapes précédentes.

```
response = personalize.create_dataset(
    name = 'datase_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'Interactions'
)

dataset_arn = response['datasetArn']
```

4. Importez vos données à l'aide d'une tâche d'importation de jeux de données avec le code suivant. Le code utilise la méthode describe_dataset_import_job pour suivre l'état de la tâche.

   Transmettez les paramètres suivants : le nom de la tâche, celui `dataset_arn` de l'étape précédente, le chemin du compartiment Amazon S3 (`s3://bucket name/folder name/ratings.csv`) dans lequel vous avez stocké les données de formation et l'ARN de votre rôle de service IAM. Vous avez créé ce rôle dans le cadre duConditions préalables pour démarrer. Amazon Personalize a besoin d'une autorisation pour accéder au compartiment. Consultez Permettre à Amazon Personalize d'accéder aux ressources Amazon S3.

```
import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation':'s3://amzn-s3-demo-bucket/filename.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

Étape 3 : Création d'une solution

Après avoir importé vos données, vous créez une solution et une version de solution comme suit. La solution contient les configurations permettant d'entraîner un modèle et une version de solution est un modèle entraîné.

1. Créez une nouvelle solution avec le code suivant. Passez les paramètres suivants : le `dataset_group_arn` précédent, le nom de la solution et l'ARN de la recette User-

Personalization-v 2 (`arn:aws:personalize:::recipe/aws-user-personalization-v2`). Stockez l'ARN de votre nouvelle solution pour une utilisation ultérieure.

```
create_solution_response = personalize.create_solution(
  name='solution name',
  recipeArn= 'arn:aws:personalize:::recipe/aws-user-personalization-v2',
  datasetGroupArn = 'dataset group arn'
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

2. Créez une version de solution avec le code suivant. Passez en paramètre le résultat `solution_arn` de l'étape précédente. Le code suivant crée une version de solution. Pendant l'entraînement, le code utilise l'DescribeSolutionVersionopération pour récupérer l'état de la version de la solution. Lorsque la formation est terminée, la méthode renvoie l'ARN de la nouvelle version de votre solution. Conservez-le en vue d'une utilisation ultérieure.

```
import time
import json

create_solution_version_response = personalize.create_solution_version(
    solutionArn = solution_arn
)

solution_version_arn = create_solution_version_response['solutionVersionArn']
print(json.dumps(create_solution_version_response, indent=2))

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_solution_version_response = personalize.describe_solution_version(
        solutionVersionArn = solution_version_arn
    )
    status = describe_solution_version_response["solutionVersion"]["status"]
    print("SolutionVersion: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

Étape 4 : Créer une campagne

Après avoir créé la version de votre solution, déployez-la dans le cadre d'une campagne Amazon Personalize. Utilisez le code suivant pour créer une campagne qui déploie la version de votre solution. Passez les paramètres suivants : le`solution_version_arn`, et le nom de la campagne. La méthode renvoie l'Amazon Resource Name (ARN) de votre nouvelle campagne. Conservez-le en vue d'une utilisation ultérieure.

```
response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn'
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

Étape 5 : Obtenir des recommandations

Une fois que vous avez créé une campagne, vous pouvez l'utiliser pour obtenir des recommandations. Le code suivant montre comment obtenir des recommandations à partir d'une campagne et imprimer l'identifiant de chaque article recommandé. Transmettez l'ARN de la campagne que vous avez créée à l'étape précédente. Pour l'ID utilisateur, vous transmettez l'ID d'un utilisateur issu des données d'entraînement, telles que`123`.

```
response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = '123',
    numResults = 10
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Commencer à utiliser les blocs-notes Amazon Personalize APIs avec Jupyter (iPython)

Pour commencer à utiliser Amazon Personalize à l'aide des blocs-notes Jupyter, clonez ou téléchargez une série de carnets situés dans le dossier getting_started du référentiel d'échantillons Amazon Personalize. Les carnets de notes vous aident à importer des données de formation, à créer une solution, à créer une campagne et à obtenir des recommandations à l'aide d'Amazon Personalize.

> **Note**
>
> Avant de commencer avec les blocs-notes, assurez-vous de créer votre environnement en suivant les étapes du fichier README.md

Mise en route (SDK pour Java 2.x)

Ce didacticiel vous explique comment effectuer le flux de travail Amazon Personalize du début à la fin avec le AWS SDK for Java 2.x.

Lorsque vous avez terminé l'exercice de mise en route, pour éviter d'encourir des frais inutiles, supprimez les ressources que vous avez créées. Pour de plus amples informations, veuillez consulter Conditions requises pour supprimer des ressources Amazon Personalize.

Pour obtenir plus d’exemples, consultez Compléter le projet Amazon Personalize.

Rubriques

- Prérequis
- Compléter le projet Amazon Personalize

Prérequis

Les étapes suivantes sont nécessaires pour suivre ce didacticiel :

- Complétez leConditions préalables pour démarrer, pour configurer les autorisations requises et créer les données d'entraînement. Vous pouvez utiliser les mêmes données sources que celles utilisées dans les Démarrer (AWS CLI) exercices Démarrer (console) ou. Si vous utilisez vos propres données sources, assurez-vous qu'elles sont formatées conformément aux prérequis.

- Configurez votre environnement SDK for Java 2.x AWS et vos informations d'identification comme indiqué dans la procédure de configuration AWS SDK for Java 2.x du manuel du développeur AWS SDK for Java 2.x .

didacticiel

Dans les étapes suivantes, vous allez configurer votre projet pour utiliser les packages Amazon Personalize et créer le SDK Amazon Personalize pour les clients Java 2.x. Ensuite, vous importez des données, vous créez et déployez une version de solution avec une campagne et vous obtenez des recommandations.

Étape 1 : configurer votre projet pour utiliser les packages Amazon Personalize

Une fois les conditions requises remplies, ajoutez les dépendances Amazon Personalize à votre fichier pom.xml et importez les packages Amazon Personalize.

1. Ajoutez les dépendances suivantes à votre fichier pom.xml. Les numéros de version les plus récents peuvent être différents de ceux de l'exemple de code.

```
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalize</artifactId>
 <version>2.16.83</version>
</dependency>
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalizeruntime</artifactId>
 <version>2.16.83</version>
</dependency>
<dependency>
 <groupId>software.amazon.awssdk</groupId>
 <artifactId>personalizeevents</artifactId>
 <version>2.16.83</version>
</dependency>
```

2. Ajoutez les instructions d'importation suivantes à votre projet.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
```

```
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
 software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
 software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// solution packages
import software.amazon.awssdk.services.personalize.model.CreateSolutionRequest;
import software.amazon.awssdk.services.personalize.model.CreateSolutionResponse;
// solution version packages
import software.amazon.awssdk.services.personalize.model.DescribeSolutionRequest;
import
 software.amazon.awssdk.services.personalize.model.CreateSolutionVersionRequest;
import
 software.amazon.awssdk.services.personalize.model.CreateSolutionVersionResponse;
import
 software.amazon.awssdk.services.personalize.model.DescribeSolutionVersionRequest;
// campaign packages
import software.amazon.awssdk.services.personalize.model.CreateCampaignRequest;
import software.amazon.awssdk.services.personalize.model.CreateCampaignResponse;
// get recommendations packages
import
 software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
 software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

Étape 2 : créer des clients Amazon Personalize

Après avoir ajouté les dépendances Amazon Personalize à votre fichier pom.xml et importé les packages requis, créez les clients Amazon Personalize suivants :

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
  .region(region)
  .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
  .region(region)
  .build();
```

Étape 3 : Importer des données

Après avoir initialisé vos clients Amazon Personalize, importez les données historiques que vous avez créées lorsque vous avez terminé leConditions préalables pour démarrer. Pour importer des données historiques dans Amazon Personalize, procédez comme suit :

1. Enregistrez le schéma Avro suivant sous forme de fichier JSON dans votre répertoire de travail. Ce schéma correspond aux colonnes du fichier CSV que vous avez créé lorsque vous avez terminé leConditions préalables pour démarrer.

   ```
   {
     "type": "record",
     "name": "Interactions",
     "namespace": "com.amazonaws.personalize.schema",
     "fields": [
         {
             "name": "USER_ID",
             "type": "string"
         },
         {
             "name": "ITEM_ID",
             "type": "string"
         },
         {
             "name": "TIMESTAMP",
             "type": "long"
         }
     ],
     "version": "1.0"
   }
   ```

2. Utilisez la `createSchema` méthode suivante pour créer un schéma dans Amazon Personalize. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de votre schéma et le chemin du fichier JSON du schéma que vous avez créé à l'étape précédente. La

méthode renvoie l'Amazon Resource Name (ARN) de votre nouveau schéma. Conservez-le en vue d'une utilisation ultérieure.

```
   public static String createSchema(PersonalizeClient personalizeClient, String
schemaName, String filePath) {

       String schema = null;
       try {
           schema = new String(Files.readAllBytes(Paths.get(filePath)));
       } catch (IOException e) {
           System.out.println(e.getMessage());
       }

       try {
           CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
                   .name(schemaName)
                   .schema(schema)
                   .build();

           String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

           System.out.println("Schema arn: " + schemaArn);

           return schemaArn;

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return "";
   }
```

3. Créez un groupe d'ensemble de données. Utilisez la `createDatasetGroup` méthode suivante pour créer un groupe de jeux de données. Transmettez les paramètres suivants : un client du service Amazon Personalize et le nom du groupe de jeux de données. La méthode renvoie l'ARN de votre nouveau groupe de données. Conservez-le en vue d'une utilisation ultérieure.

```
   public static String createDatasetGroup(PersonalizeClient personalizeClient,
String datasetGroupName) {

       try {
```

```
            CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
                    .name(datasetGroupName)
                    .build();
            return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

4. Créez un jeu de données sur les interactions entre les articles. Utilisez la `createDataset` méthode suivante pour créer un jeu de données d'interactions entre articles. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de votre ensemble de données, l'ARN de votre schéma, l'ARN de votre groupe de jeux de données et le type `Interactions` de jeu de données. La méthode renvoie l'ARN de votre nouvel ensemble de données. Conservez-le en vue d'une utilisation ultérieure.

```
    public static String createDataset(PersonalizeClient personalizeClient,
            String datasetName,
            String datasetGroupArn,
            String datasetType,
            String schemaArn) {
        try {
            CreateDatasetRequest request = CreateDatasetRequest.builder()
                    .name(datasetName)
                    .datasetGroupArn(datasetGroupArn)
                    .datasetType(datasetType)
                    .schemaArn(schemaArn)
                    .build();

            String datasetArn = personalizeClient.createDataset(request)
                    .datasetArn();
            System.out.println("Dataset " + datasetName + " created.");
            return datasetArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
```

```
    }
```

5. Importez vos données à l'aide d'une tâche d'importation de jeux de données. Utilisez la `createPersonalizeDatasetImportJob` méthode suivante pour créer une tâche d'importation de jeux de données.

   Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de la tâche, l'ARN de votre ensemble de données d'interactions avec les articles, le chemin du compartiment Amazon S3 (`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`) dans lequel vous avez stocké les données de formation et l'ARN de votre rôle de service (vous avez créé ce rôle dans le cadre duConditions préalables pour démarrer). La méthode renvoie l'ARN de votre tâche d'importation de jeu de données. Stockez-le éventuellement pour une utilisation ultérieure.

```
   public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
           String jobName,
           String datasetArn,
           String s3BucketPath,
           String roleArn) {

       long waitInMilliseconds = 60 * 1000;
       String status;
       String datasetImportJobArn;

       try {
           DataSource importDataSource = DataSource.builder()
                   .dataLocation(s3BucketPath)
                   .build();

           CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
                   .datasetArn(datasetArn)
                   .dataSource(importDataSource)
                   .jobName(jobName)
                   .roleArn(roleArn)
                   .build();

           datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
                   .datasetImportJobArn();
           DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
```

```
                .datasetImportJobArn(datasetImportJobArn)
                .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetImportJob datasetImportJob = personalizeClient
                    .describeDatasetImportJob(describeDatasetImportJobRequest)
                    .datasetImportJob();

            status = datasetImportJob.status();
            System.out.println("Dataset import job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return datasetImportJobArn;

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

Étape 4 : Création d'une solution

Après avoir importé vos données, vous créez une solution et une version de solution comme suit. La solution contient les configurations permettant d'entraîner un modèle et une version de solution est un modèle entraîné.

1. Créez une nouvelle solution à l'aide de la `createPersonalizeSolution` méthode suivante. Transmettez les paramètres suivants : un client du service Amazon Personalize, vos groupes de données Amazon Resource Name (ARN), le nom de la solution et l'ARN des User-Personalization-

v 2 recettes (`arn:aws:personalize:::recipe/aws-user-personalization-v2`). La méthode renvoie l'ARN de votre nouvelle solution. Conservez-le en vue d'une utilisation ultérieure.

```
   public static String createPersonalizeSolution(PersonalizeClient
personalizeClient,
           String datasetGroupArn,
           String solutionName,
           String recipeArn) {

       try {
           CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
                   .name(solutionName)
                   .datasetGroupArn(datasetGroupArn)
                   .recipeArn(recipeArn)
                   .build();

           CreateSolutionResponse solutionResponse =
personalizeClient.createSolution(solutionRequest);
           return solutionResponse.solutionArn();

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return "";
   }
```

2. Créez une version de solution à l'aide de la `createPersonalizeSolutionVersion` méthode suivante. Passez en paramètre l'ARN de la solution à l'étape précédente. Le code suivant vérifie d'abord si votre solution est prête, puis crée une version de la solution. Pendant l'entraînement, le code utilise l'DescribeSolutionVersionopération pour récupérer l'état de la version de la solution. Lorsque la formation est terminée, la méthode renvoie l'ARN de la nouvelle version de votre solution. Conservez-le en vue d'une utilisation ultérieure.

```
   public static String createPersonalizeSolutionVersion(PersonalizeClient
personalizeClient, String solutionArn) {
       long maxTime = 0;
       long waitInMilliseconds = 30 * 1000; // 30 seconds
       String solutionStatus = "";
       String solutionVersionStatus = "";
       String solutionVersionArn = "";
```

```
        try {
            DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
                    .solutionArn(solutionArn)
                    .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            // Wait until solution is active.
            while (Instant.now().getEpochSecond() < maxTime) {

                solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
                System.out.println("Solution status: " + solutionStatus);

                if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }

            if (solutionStatus.equals("ACTIVE")) {

                CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
                        .solutionArn(solutionArn)
                        .build();

                CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient
                        .createSolutionVersion(createSolutionVersionRequest);
                solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

                System.out.println("Solution version ARN: " + solutionVersionArn);

                DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
                        .solutionVersionArn(solutionVersionArn)
```

```
                        .build();

                while (Instant.now().getEpochSecond() < maxTime) {

                    solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
                            .solutionVersion().status();
                    System.out.println("Solution version status: " +
solutionVersionStatus);

                    if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                        break;
                    }
                    try {
                        Thread.sleep(waitInMilliseconds);
                    } catch (InterruptedException e) {
                        System.out.println(e.getMessage());
                    }
                }
                return solutionVersionArn;
            }
        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

Pour de plus amples informations, veuillez consulter Création manuelle d'une version de solution. Lorsque vous créez une version de solution, vous pouvez évaluer ses performances avant de continuer. Pour de plus amples informations, veuillez consulter Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques.

Étape 5 : créer une campagne

Après avoir formé et évalué la version de votre solution, déployez-la dans le cadre d'une campagne Amazon Personalize. Utilisez la `createPersonalCampaign` méthode suivante pour déployer une version de solution. Transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de ressource Amazon (ARN) de la version de solution que vous avez créée à l'étape

précédente et le nom de la campagne. La méthode renvoie l'ARN de votre nouvelle campagne. Conservez-le en vue d'une utilisation ultérieure.

```
public static String createPersonalCompaign(PersonalizeClient personalizeClient, String
 solutionVersionArn, String name) {

    try {
        CreateCampaignRequest createCampaignRequest = CreateCampaignRequest.builder()
            .minProvisionedTPS(1)
            .solutionVersionArn(solutionVersionArn)
            .name(name)
            .build();

        CreateCampaignResponse campaignResponse =
 personalizeClient.createCampaign(createCampaignRequest);
        System.out.println("The campaign ARN is "+campaignResponse.campaignArn());
        return campaignResponse.campaignArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Pour plus d'informations sur les campagnes Amazon Personalize, consultezDéploiement d'une version de la solution Amazon Personalize avec une campagne.

Étape 6 : Obtenir des recommandations

Après avoir créé une campagne, vous l'utilisez pour obtenir des recommandations. Utilisez la `getRecs` méthode suivante pour obtenir des recommandations pour un utilisateur. Transmettez comme paramètres un client d'exécution Amazon Personalize, le nom de ressource Amazon (ARN) de la campagne que vous avez créée à l'étape précédente et un ID utilisateur (par exemple`123`) issu des données historiques que vous avez importées. La méthode imprime la liste des éléments recommandés à l'écran.

```
    public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
 String campaignArn, String userId) {

        try {
            GetRecommendationsRequest recommendationsRequest =
 GetRecommendationsRequest.builder()
```

```
                    .campaignArn(campaignArn)
                    .numResults(20)
                    .userId(userId)
                    .build();

            GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
                    .getRecommendations(recommendationsRequest);
            List<PredictedItem> items = recommendationsResponse.itemList();
            for (PredictedItem item : items) {
                System.out.println("Item Id is : " + item.itemId());
                System.out.println("Item score is : " + item.score());
            }

        } catch (AwsServiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

Compléter le projet Amazon Personalize

Pour un all-in-one projet qui explique comment terminer le flux de travail Amazon Personalize avec le SDK pour Java 2.x, consultez l'application Amazon-Personalize-Java sur. GitHub Ce projet inclut la formation de plusieurs versions de solutions avec différentes recettes et l'enregistrement des événements liés à l' PutEvents opération.

Pour des exemples supplémentaires, consultez le code trouvé dans le dossier de personnalisation du référentiel d'exemples du AWS SDK.

Adapter votre cas d'utilisation aux ressources Amazon Personalize

Les recommandations d'Amazon Personalize peuvent répondre aux cas d'utilisation suivants :

- Génération de recommandations personnalisées pour un utilisateur
- Recommander des articles similaires ou connexes
- Recommander des articles tendance ou populaires
- Recommander les meilleures actions suivantes à un utilisateur (uniquement avec des ressources personnalisées)
- Réorganisation par pertinence (uniquement avec des ressources personnalisées)
- Génération de segments d'utilisateurs (uniquement avec des ressources personnalisées)

Amazon Personalize propose des ressources basées sur le domaine et des ressources personnalisées configurées pour ces cas d'utilisation. Vous commencez par créer un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé :

- Avec un groupe de jeux de données de domaine, vous créez des ressources préconfigurées et optimisées pour les domaines VIDEO_ON_DEMAND ou ECOMMERCE.

  Si vous possédez une application de streaming vidéo ou de commerce électronique, nous vous recommandons de commencer par un groupe de jeux de données de domaine. Vous pouvez toujours ajouter des ressources personnalisées, telles que des solutions et des versions de solutions conçues pour des cas d'utilisation personnalisés. Et vous pouvez toujours utiliser des ressources personnalisées pour obtenir des recommandations par lots. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine.
- Avec un groupe de jeux de données personnalisé, vous choisissez une recette qui correspond à votre cas d'utilisation. Vous formez et déployez ensuite uniquement des solutions et des versions de solutions configurables (modèles de recommandation Amazon Personalize entraînés). Lorsque vous êtes prêt, vous pouvez déployer la version de la solution dans le cadre d'une campagne pour obtenir des recommandations en temps réel. Vous pouvez également obtenir des recommandations par lots sans campagne.

Si vous n'avez pas d'application de streaming vidéo ou de commerce électronique, nous vous recommandons de créer un groupe de jeux de données personnalisé. Sinon, commencez par un groupe de jeux de données de domaine et ajoutez des ressources personnalisées si nécessaire.

Les sections suivantes fournissent des informations détaillées sur les cas d'utilisation et les recettes personnalisées disponibles dans Amazon Personalize. Lorsque vous associez votre cas d'utilisation à une ressource Amazon Personalize, notez ses exigences en matière de données. Une fois que vous avez choisi un cas d'utilisation ou une recette, ces informations peuvent vous aider à préparer vos données dansPréparation des données de formation pour Amazon Personalize.

Rubriques

- Cas d'utilisation et fonctionnalités des recettes
- Choix d'un cas d'utilisation
- Choisir une recette

Cas d'utilisation et fonctionnalités des recettes

Avec certains cas d'utilisation et certaines recettes, Amazon Personalize utilise les fonctionnalités suivantes pour générer des recommandations plus pertinentes et améliorer la découverte des articles et l'engagement.

Rubriques

- Personnalisation en temps réel
- Exploration
- Mises à jour automatiques

Personnalisation en temps réel

Avec certains cas d'utilisation et certaines recettes, Amazon Personalize utilise la personnalisation en temps réel pour mettre à jour et adapter les recommandations en fonction de l'évolution des intérêts de l'utilisateur. Il met à jour les recommandations destinées à un utilisateur lorsque vous enregistrez ses interactions avec des éléments ou des actions présents lors de la dernière formation complète. Vous enregistrez ces interactions avec un outil de suivi des événements et l'PutEventsopération ou, pour les interactions avec des actions, l' PutActionInteractions opération.

Pour plus d'informations sur l'enregistrement d'événements, consultezEnregistrement des événements en temps réel pour influencer les recommandations. Pour plus d'informations sur l'influence des nouvelles données sur les recommandations en temps réel, y compris la personnalisation en temps réel, voirMise à jour des données dans les ensembles de données après l'entraînement.

Les exemples d'utilisation et les recettes suivants prennent en charge la personnalisation en temps réel :

- Recommandé pour vous (cas d'utilisation du commerce électronique)
- Les meilleurs choix pour vous (cas d'utilisation de VIDEO_ON_DEMAND)
- Recette de personnalisation par l'utilisateur V2
- Recette de personnalisation par l'utilisateur
- Recette de classement V2 personnalisée
- Recette de classement personnalisé
- Next-Best-Action recette

Exploration

Pour certains cas d'utilisation de domaines et certaines recettes personnalisées, Amazon Personalize utilise l'exploration lorsqu'il recommande des articles. Dans le cadre de l'exploration, les recommandations incluent certains éléments ou actions qui sont généralement moins susceptibles d'être recommandés à l'utilisateur, tels que de nouveaux éléments ou actions, des éléments ou des actions avec peu d'interactions, ou des éléments ou actions moins pertinents pour l'utilisateur en raison de son comportement antérieur. Cela améliore la découverte d'articles et l'engagement lorsque votre catalogue évolue rapidement ou lorsque de nouveaux articles, tels que des articles de presse ou des promotions, sont plus pertinents pour les utilisateurs parce qu'ils sont récents.

Configuration de l'exploration

Si vous utilisez la recette User-Personalization-v 2, Amazon Personalize gère la configuration de l'exploration pour vous et les éléments inclus dans le cadre de l'exploration figurent `Reason` dans la réponse de recommandation. `Exploration` Pour vous assurer que les nouveaux articles sont inclus dans les recommandations, vous pouvez utiliser un filtre de promotion pour promouvoir les nouveaux articles en fonction de l'horodatage de création. Pour plus d'informations sur les promotions, consultezPromouvoir des articles dans le cadre de recommandations en temps réel.

Pour tous les autres cas d'utilisation ou recettes utilisant l'exploration, lorsque vous créez un recommandateur ou une campagne personnalisée, ou lorsque vous créez une tâche d'inférence par lots (ressources personnalisées), vous pouvez configurer l'exploration avec les champs suivants :

- Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
- Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

  Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

Cas d'utilisation et recettes utilisant l'exploration

Pour plus d'informations sur chaque cas d'utilisation ou recette faisant appel à l'exploration, consultez les rubriques suivantes :

- Recommandé pour vous (cas d'utilisation du commerce électronique)
- Les meilleurs choix pour vous (cas d'utilisation de VIDEO_ON_DEMAND)
- Recette de personnalisation par l'utilisateur V2
- Recette de personnalisation par l'utilisateur
- Next-Best-Action recette

Mises à jour automatiques

Pour certains cas d'utilisation et certaines recettes personnalisées, Amazon Personalize met automatiquement à jour la version de votre recommandation ou de votre solution afin de prendre en compte de nouveaux éléments ou de nouvelles actions pour les recommandations. Les mises à jour automatiques sont gratuites. Pour une liste des cas d'utilisation et des recettes avec mises à jour automatiques, voirCas d'utilisation du domaine et recettes personnalisées avec mises à jour automatiques.

Les mises à jour automatiques fonctionnent comme suit :

- La date à laquelle Amazon Personalize met automatiquement à jour la version de votre solution ou votre outil de recommandation dépend de la manière dont vous recevez les recommandations :
  - Pour des recommandations en temps réel, Amazon Personalize met à jour la version de la solution ou le logiciel de recommandation toutes les deux heures.
  - Pour les recommandations d'articles par lots, lorsque vous créez une tâche d'inférence par lots et que vous spécifiez la dernière version de solution entièrement formée pour votre solution, Amazon Personalize met automatiquement à jour la version de la solution afin de prendre en compte les nouveaux articles lors de l'exploration. Si vous ne spécifiez pas la dernière version de la solution, aucune mise à jour n'est effectuée.
- À chaque mise à jour, Amazon Personalize commence à inclure de nouveaux articles dans les recommandations d'utilisationExploration. Lorsque vous envisagez un nouvel article ou une nouvelle action, Amazon Personalize prend en compte toutes les métadonnées associées à l'article. Toutefois, ces données n'auront un effet plus important sur les recommandations qu'une fois que vous aurez enregistré les interactions relatives à l'élément et que vous vous serez complètement réentraîné.
- Pour qu'une mise à jour ait lieu, vous devez fournir de nouvelles données d'action, d'élément ou d'interaction depuis la dernière mise à jour automatique ou le dernier réentraînement.
- Amazon Personalize prend en compte les nouveaux articles jusqu'à ce que vous en importiez 750 000. Il s'agit du nombre maximum d'éléments pris en compte pendant l'entraînement.

Directives et exigences supplémentaires pour les ressources personnalisées

Si vous utilisez des ressources personnalisées, voici les directives et les exigences relatives aux mises à jour automatiques :

- La version de votre solution doit être déployée dans le cadre d'une campagne. Votre campagne utilise automatiquement la version mise à jour de la solution.
- Les mises à jour automatiques ne sont pas la même chose que l'entraînement automatique. Une mise à jour automatique ne crée pas une version complètement nouvelle de la solution. Et le modèle ne tire aucune leçon de vos dernières données. Pour maintenir votre solution, votre fréquence d'entraînement automatique doit toujours être d'au moins une fois par semaine.
- Une fois que votre solution a automatiquement créé une nouvelle version de solution ou que vous en avez créé une nouvelle manuellement, Amazon Personalize ne met pas automatiquement à jour les anciennes versions de la solution, même si vous les avez déployées dans le cadre d'une campagne.
- Si toutes les deux heures ne sont pas assez fréquentes, grâce à la personnalisation par l'utilisateur, vous pouvez créer manuellement une version de solution `trainingMode` configurée `UPDATE` pour inclure ces nouveaux éléments dans les recommandations. N'oubliez pas qu'Amazon Personalize met automatiquement à jour uniquement la dernière version complète de votre solution. La version de la solution mise à jour manuellement ne sera pas automatiquement mise à jour à l'avenir. Si votre solution utilise l'entraînement automatique, les mises à jour automatiques reprendront pour la prochaine version de la solution. Si ce n'est pas le cas, créez manuellement une nouvelle solution avec le mode formation configuré `FULL` et déployez-la dans le cadre d'une campagne.

Cas d'utilisation du domaine et recettes personnalisées avec mises à jour automatiques

Pour plus d'informations sur chaque cas d'utilisation ou recette comportant des mises à jour automatiques, consultez les rubriques suivantes :

- Recommandé pour vous (cas d'utilisation du commerce électronique)
- Les meilleurs choix pour vous (cas d'utilisation de VIDEO_ON_DEMAND)
- Recette de personnalisation par l'utilisateur V2
- Recette de personnalisation par l'utilisateur
- Next-Best-Action recette

Choix d'un cas d'utilisation

Lorsque vous créez une recommandation dans un groupe de jeux de données de domaine, vous spécifiez un cas d'utilisation. Amazon Personalize forme les modèles supportant le recommandeur avec les configurations les mieux adaptées au cas d'utilisation. Chaque domaine a des cas d'utilisation différents. Par exemple, si vous spécifiez VIDEO_ON_DEMAND pour votre groupe de jeux de données de domaine, seuls les cas d'utilisation VIDEO_ON_DEMAND sont disponibles. Chaque cas d'utilisation comporte des exigences différentes pour obtenir des recommandations. Certains cas d'utilisation nécessitent des types d'événements spécifiques. Vous êtes libre d'inclure des types d'événements supplémentaires.

Pour tous les cas d'utilisation, vos données d'interactions doivent présenter les éléments suivants :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

Rubriques

- Cas d'utilisation de VIDEO_ON_DEMAND
- Cas d'utilisation du commerce électronique

Cas d'utilisation de VIDEO_ON_DEMAND

Les sections suivantes répertorient les exigences et le nom de ressource Amazon (ARN) pour chaque cas d'utilisation de VIDEO_ON_DEMAND. Pour tous les cas d'utilisation, vos données d'interactions doivent présenter les éléments suivants :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

> **Note**
>
> Si vous utilisez l'CreateRecommenderAPI, indiquez l'ARN répertorié ici pour l'ARN de recette.

Rubriques

- Parce que tu as regardé X
- Plutôt X
- Les plus populaires
- À la mode en ce moment
- Les meilleurs choix pour vous

Parce que tu as regardé X

Obtenez des recommandations pour les vidéos que d'autres utilisateurs ont également regardées sur la base d'une vidéo que vous spécifiez. Dans ce cas d'utilisation, Amazon Personalize filtre automatiquement les vidéos regardées par l'utilisateur en fonction de l'userID que vous spécifiez et des `Watch` événements. Si vous appliquez votre propre filtre, celui-ci est appliqué une fois que les vidéos regardées par l'utilisateur ont été filtrées.

Lors du filtrage, Amazon Personalize prend en compte au maximum 100 interactions entre articles par utilisateur et par type d'événement. Cela s'applique à tous les filtres automatiques ou personnalisés. Vous pouvez utiliser la console Service Quotas pour demander une augmentation de cette limite. Pour plus d'informations, consultez la section Demander une augmentation de quota du Guide de l'utilisateur du Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-vod-because-you-watched-x`
- GetRecommendations Exigences relatives à l'API :

  `userId` : Obligatoire

`itemId` : Obligatoire

- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `Watch` événements.

Plutôt X

Obtenez des recommandations pour des vidéos similaires à une vidéo que vous spécifiez. Dans ce cas d'utilisation, Amazon Personalize filtre automatiquement les vidéos regardées par l'utilisateur en fonction de l'userID que vous spécifiez et `Watch` des événements. Si vous appliquez votre propre filtre, celui-ci est appliqué une fois que les vidéos regardées par l'utilisateur ont été filtrées.

Lors du filtrage, Amazon Personalize prend en compte au maximum 100 interactions entre articles par utilisateur et par type d'événement. Cela s'applique à tous les filtres automatiques ou personnalisés. Vous pouvez utiliser la console Service Quotas pour demander une augmentation de cette limite. Pour plus d'informations, consultez la section Demander une augmentation de quota du Guide de l'utilisateur du Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-vod-more-like-x`
- GetRecommendations Exigences relatives à l'API :

  `userId` : Obligatoire

  `itemId` : Obligatoire
- Ensembles de données utilisés lors de l'entraînement :
  - Interactions (obligatoire)
  - Objets (obligatoires)
- Nombre d'événements requis : au moins 1 000 événements de tout type.
- Types d'événements recommandés : `Watch` et `Click` événements.

Les plus populaires

Obtenez des recommandations pour les vidéos qui ont été regardées par le plus grand nombre d'utilisateurs.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-vod-most-popular`
- GetRecommendationsexigences :

  `userId` : Obligatoire

  `itemId` : Non utilisé
- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `Watch` événements.

À la mode en ce moment

Obtenez des recommandations pour les vidéos actuellement à la mode. Les vidéos tendance sont des articles qui gagnent rapidement en popularité auprès de vos utilisateurs. Toutes les deux heures, Amazon Personalize évalue automatiquement les données de vos interactions et identifie les articles tendance.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-vod-trending-now`
- GetRecommendations Exigences relatives à l'API :

  `userId`: obligatoire uniquement si vous filtrez par CurrentUser ou par éléments avec lesquels un utilisateur a interagi

  `itemId` : Non utilisé
- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Nombre d'événements requis : au moins 1 000 événements de tout type.

Les meilleurs choix pour vous

Obtenez des recommandations de contenu personnalisées pour un utilisateur que vous spécifiez. Dans ce cas d'utilisation, Amazon Personalize filtre automatiquement les vidéos regardées par

l'utilisateur en fonction de l'userID que vous spécifiez et `Watch` des événements. Si vous appliquez votre propre filtre, celui-ci est appliqué une fois que les vidéos regardées par l'utilisateur ont été filtrées.

Lors du filtrage, Amazon Personalize prend en compte au maximum 100 interactions entre articles par utilisateur et par type d'événement. Cela s'applique à tous les filtres automatiques ou personnalisés. Vous pouvez utiliser la console Service Quotas pour demander une augmentation de cette limite. Pour plus d'informations, consultez la section Demander une augmentation de quota du Guide de l'utilisateur du Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

Lorsque vous recommandez des articles, ce cas d'utilisation utilise real-time-personalizationet explore. Et il utilise des mises à jour automatiques pour prendre en compte les nouveaux éléments et les recommander.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-vod-top-picks`
- GetRecommendations exigences :

  `userId` : Obligatoire

  `itemId` : Non utilisé
- Ensembles de données utilisés lors de l'entraînement :
  - Interactions (obligatoire)
  - Objets (facultatif)
  - Utilisateurs (facultatif)
- Nombre d'événements requis : au moins 1 000 événements.
- Types d'événements recommandés : `Click` et `Watch` événements.
- Paramètres de configuration de l'exploration : Lorsque vous créez un recommandateur, vous pouvez configurer l'exploration avec les éléments suivants.
  - Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).

- Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

  Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

Cas d'utilisation du commerce électronique

Les sections suivantes répertorient les exigences et le nom de ressource Amazon (ARN) pour chaque cas d'utilisation du commerce électronique. Pour tous les cas d'utilisation, vos données d'interactions doivent présenter les éléments suivants :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

> **Note**
>
> Si vous utilisez l'CreateRecommenderAPI, indiquez l'ARN répertorié ici pour l'ARN de recette.

Rubriques

- Les plus regardés

- Les meilleures ventes
- Souvent achetés ensemble
- Les clients ayant consulté X ont également consulté
- Recommandé pour vous

Les plus regardés

Obtenez des recommandations pour des articles populaires en fonction du nombre de fois que vos clients ont consulté un article.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-views`
- GetRecommendations exigences :

  `userId` : Obligatoire

  `itemId` : Non utilisé

  `inputList` : ND
- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `View` événements.

Les meilleures ventes

Obtenez des recommandations pour des articles populaires en fonction du nombre de fois que vos clients ont acheté un article.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-purchases`
- GetRecommendations exigences :

  `userId` : Obligatoire

  `itemId` : Non utilisé

  `inputList` : ND

- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `Purchase` événements.

Souvent achetés ensemble

Obtenez des recommandations pour les articles que les clients achètent fréquemment en même temps qu'un article que vous spécifiez.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-ecomm-frequently-bought-together`
- GetRecommendations exigences :

  `userId`: obligatoire uniquement si vous filtrez par CurrentUser

  `itemId` : Obligatoire

  `inputList` : ND
- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `Purchase` événements.

Les clients ayant consulté X ont également consulté

Obtenez des recommandations pour les articles que les clients ont également consultés en fonction d'un article que vous spécifiez. Dans ce cas d'utilisation, Amazon Personalize filtre automatiquement les articles achetés par l'utilisateur en fonction de l'userID que vous spécifiez et `Purchase` des événements. Si vous appliquez votre propre filtre, celui-ci est appliqué une fois que les articles déjà achetés par l'utilisateur ont été filtrés.

Lors du filtrage, Amazon Personalize prend en compte au maximum 100 interactions entre articles par utilisateur et par type d'événement. Cela s'applique à tous les filtres automatiques ou personnalisés. Vous pouvez utiliser la console Service Quotas pour demander une augmentation de cette limite. Pour plus d'informations, consultez la section Demander une augmentation de quota du Guide de l'utilisateur du Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de

l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-ecomm-customers-who-viewed-x-also-viewed`
- GetRecommendations exigences :

  `userId` : Obligatoire

  `itemId` : Obligatoire

  `inputList` : ND
- Ensembles de données utilisés lors de l'entraînement : ensemble de données sur les interactions entre éléments uniquement (obligatoire)
- Types d'événements requis : au moins 1 000 `View` événements.
- Types d'événements recommandés : `Purchase` événements.

Recommandé pour vous

Obtenez des recommandations personnalisées pour les articles en fonction de l'utilisateur que vous spécifiez. Dans ce cas d'utilisation, Amazon Personalize filtre automatiquement les articles achetés par l'utilisateur en fonction de l'userID que vous spécifiez et `Purchase` des événements. Si vous appliquez votre propre filtre, celui-ci est appliqué une fois que les articles déjà achetés par l'utilisateur ont été filtrés.

Lors du filtrage, Amazon Personalize prend en compte au maximum 100 interactions entre articles par utilisateur et par type d'événement. Cela s'applique à tous les filtres automatiques ou personnalisés. Vous pouvez utiliser la console Service Quotas pour demander une augmentation de cette limite. Pour plus d'informations, consultez la section Demander une augmentation de quota du Guide de l'utilisateur du Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

Lorsque vous recommandez des articles, ce cas d'utilisation utilise real-time-personalizationet explore. Et il utilise des mises à jour automatiques pour prendre en compte les nouveaux éléments et les recommander.

- ARN de la recette : `arn:aws:personalize:::recipe/aws-ecomm-recommended-for-you`
- GetRecommendations exigences :

  `userId` : Obligatoire

  `itemId` : Non utilisé

  `inputList` : ND
- Ensembles de données utilisés lors de l'entraînement :
  - Interactions (obligatoire)
  - Objets (facultatif)
  - Utilisateurs (facultatif)
- Nombre d'événements requis : au moins 1 000 événements.
- Types d'événements recommandés : `View` et `Purchase` événements.
- Paramètres de configuration de l'exploration : Lorsque vous créez un recommandateur, vous pouvez configurer l'exploration avec les éléments suivants.
  - Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
  - Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

    Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

Choisir une recette

Lorsque vous créez une solution personnalisée, vous spécifiez une recette et configurez les paramètres d'entraînement. Les Recipes sont des algorithmes Amazon Personalize conçus pour des cas d'utilisation spécifiques. Amazon Personalize propose des recettes, basées sur des cas d'utilisation courants, pour les modèles de formation. Lorsque vous créez une version de solution pour la solution, Amazon Personalize entraîne les modèles qui soutiennent la version de la solution en fonction de la recette et de la configuration de formation.

Les recettes Amazon Personalize utilisent les éléments suivants pendant la formation :

- Attributs prédéfinis de vos données
- Transformations de fonction prédéfinies
- Algorithmes prédéfinis
- Réglage des paramètres initiaux des algorithmes

Pour optimiser votre modèle, vous pouvez remplacer bon nombre de ces paramètres lorsque vous créez une solution. Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Rubriques

- Amazon Personalize les types de recettes par cas d'utilisation
- Amazon Personalize : recettes
- Afficher les recettes Amazon Personalize disponibles
- Recette de personnalisation par l'utilisateur V2
- Recette de personnalisation par l'utilisateur
- Recette Trending-Now
- Recette Popularity-Count
- Recette de classement V2 personnalisée
- Recette de classement personnalisé
- Recette de similarité sémantique
- Recette d'articles similaires
- Recette SIMS
- Next-Best-Action recette
- Recette Item-Affinity

- Item-Attribute-Affinity recette
- Recettes HRNN traditionnelles

Amazon Personalize les types de recettes par cas d'utilisation

Pour choisir votre recette, choisissez d'abord votre cas d'utilisation parmi les suivants et notez le type de recette correspondant.

- Recommander des articles aux utilisateurs (recettes USER_PERSONALIZATION)

  Pour fournir des recommandations personnalisées à vos utilisateurs, entraînez votre modèle à l'aide d'une recette USER_PERSONALIZATION. Les recommandations personnalisées contribuent à améliorer l'engagement et la conversion.
- Classement des objets pour un utilisateur (recettes PERSONALIZED_RANKING)

  Pour personnaliser l'ordre des listes sélectionnées ou des résultats de recherche pour vos utilisateurs, entraînez votre modèle à l'aide d'une recette PERSONALIZED_RANKING. Les recettes PERSONALIZED_RANKING créent une liste personnalisée en reclassant une collection d'éléments d'entrée en fonction du niveau d'intérêt prévu pour un utilisateur donné. Les listes personnalisées améliorent l'expérience client et renforcent la fidélité et l'engagement des clients.
- Recommandation d'articles tendance ou populaires (recettes POPULAR_ITEMS)

  Pour recommander des articles tendance ou populaires, utilisez une recette POPULAR_ITEMS. Vous pouvez utiliser un POPULAR_ITEMS si vos clients attachent une grande importance à ce avec quoi les autres utilisateurs interagissent. Les utilisations courantes incluent la recommandation de contenu viral sur les réseaux sociaux, d'articles de dernière minute ou de vidéos sportives récentes.
- Recommander des articles similaires (recettes RELATED_ITEMS)

  Pour recommander des articles similaires, tels que des articles fréquemment achetés ensemble ou des films que d'autres utilisateurs ont également regardés, vous devez utiliser une recette RELATED_ITEMS. La recommandation d'articles similaires peut aider vos clients à découvrir des articles et à augmenter le taux de conversion des utilisateurs.
- Recommander la meilleure action suivante (recettes PERSONALIZED_ACTIONS)

  Pour recommander la prochaine meilleure action à vos utilisateurs en temps réel, comme l'inscription à votre programme de fidélité ou la demande d'une carte de crédit, vous devez utiliser

une recette PERSONALIZED_ACTIONS. En recommandant la meilleure action suivante, vous pouvez fidéliser vos clients, générer plus de revenus et améliorer l'expérience de vos utilisateurs.

- Obtenir des segments d'utilisateurs (recettes USER_SEGMENTATION)

  Pour obtenir des segments d'utilisateurs basés sur les données d'entrée des articles, tels que les utilisateurs susceptibles d'interagir avec des éléments dotés d'un certain attribut, vous devez utiliser une recette USER_SEGMENTATION. L'obtention de segments d'utilisateurs peut vous aider à créer des campagnes marketing avancées qui font la promotion de différents articles auprès de différents segments d'utilisateurs en fonction de la probabilité qu'ils entreprennent une action.

Amazon Personalize : recettes

Amazon Personalize propose les types de recettes suivants. Outre les différences comportementales, chaque type a des exigences spécifiques pour obtenir des recommandations, comme illustré dans le tableau suivant.

Type de formule	Recettes	API	Exigences relatives à l'API
USER_PERSONALIZATION	Personnalisation de l'utilisateur-v2 Personnalisation par l'utilisateur Recette HRNN (ancienne) Recette de métadonnées HRNN (ancienne) Recette HRNN-ColdStart (ancienne)	GetRecommendations	`userId` : Obligatoire `itemId` : Non utilisé `inputList` : ND
ARTICLES POPULAIRES	Tendance actuelle Comptage de popularité	GetRecommendations	`userId`: obligatoire

Type de formule	Recettes	API	Exigences relatives à l'API
			uniquement si vous appliquez un filtre qui l'exige `itemId` : Non utilisé `inputList` : ND
PERSONALIZED-RANKING	Ranking-V2 personnalisé Personalized-Ranking	GetPersonalizedRanking	`userId` : Obligatoire `itemId` : ND `inputList` : Liste d'ID d'éléments
RELATED_ITEMS	Objets similaires SIMS Recette de similarité sémantique	GetRecommendations	`userId`: obligatoire uniquement si vous appliquez un filtre qui l'exige `itemId` : Obligatoire `inputList` : ND

Type de formule	Recettes	API	Exigences relatives à l'API
ACTIONS_PERSONNALISÉES	Meilleure action suivante	GetActionRecommendations	`userId` : Obligatoire `actionId` : Non utilisé `itemId` : Non utilisé `inputList` : ND
SEGMENTATION DES UTILISATEURS	Affinité entre les objets Affinité entre les attributs et les objets	CreateBatchSegmentJob	Pour les exigences relatives au flux de travail par lots, voirObtenir des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots.

Afficher les recettes Amazon Personalize disponibles

Pour afficher la liste des recettes disponibles :

- Dans la console Amazon Personalize, choisissez un groupe de jeux de données. Dans le volet de navigation, choisissez Solutions and recipes (Solutions et recettes), puis choisissez l'onglet Recipes (Recettes).
- Avec le AWS SDK pour Python (Boto3), appelez l'ListRecipesAPI.

- À l'aide du AWS CLI, utilisez la commande suivante.

```
aws personalize list-recipes
```

Pour obtenir des informations sur une recette utilisant le SDK pour Python (Boto3), appelez l'API. DescribeRecipe Pour obtenir des informations sur une recette à l'aide de AWS CLI, utilisez la commande suivante.

```
aws personalize describe-recipe --recipe-arn recipe_arn
```

Recette de personnalisation par l'utilisateur V2

La recette User-Personalization-v 2 (aws-user-personalization-v2) recommande les articles avec lesquels un utilisateur va interagir en fonction de ses préférences. Par exemple, vous pouvez utiliser User-Personalization-v 2 pour générer des recommandations de films personnalisées pour une application de streaming ou des recommandations de produits personnalisées pour une application de vente au détail. D'autres cas d'utilisation incluent la génération de recommandations en temps réel pour un site d'actualités ou des recommandations par lots pour une campagne marketing personnalisée.

User-Personalization-v2 peuvent s'entraîner sur jusqu'à 5 millions d'objets à partir des interactions entre objets et des ensembles de données d'objets. Et il génère des recommandations plus pertinentes avec une latence inférieure àPersonnalisation par l'utilisateur.

Étant donné que User-Personalization-v 2 recommande les éléments les plus pertinents aux utilisateurs en fonction de vos données, il recommande plus fréquemment les éléments existants contenant des données d'interactions. Pour vous assurer que les recommandations incluent de nouveaux articles, vous pouvez utiliser une promotion qui inclut certains articles en fonction de l'horodatage de création. Pour plus d'informations sur les promotions, consultezPromouvoir des articles dans le cadre de recommandations en temps réel.

Cette recette utilise une architecture basée sur un transformateur pour entraîner un modèle qui apprend le contexte et suit les relations et les modèles dans vos données. Les transformateurs sont un type d'architecture de réseau neuronal qui transforme ou modifie une séquence d'entrée en séquence de sortie. Pour Amazon Personalize, la séquence de saisie est l'historique des interactions d'un utilisateur avec les articles dans vos données. La séquence de sortie correspond à leurs recommandations personnalisées. Pour plus d'informations sur les transformateurs, voir Que sont les transformateurs dans l'intelligence artificielle ? dans le AWS Cloud Computing Concepts Hub.

User-Personalization-v2 utilise un modèle de tarification différent de celui des autres recettes. Pour plus d'informations sur la tarification, consultez la section Tarification Amazon Personalize.

Rubriques

- Caractéristiques de la recette
- Ensembles de données obligatoires et facultatifs
- Propriétés et hyperparamètres

Caractéristiques de la recette

User-Personalization-v2 utilise les fonctionnalités de recette Amazon Personalize suivantes pour générer des recommandations d'articles :

- Personnalisation en temps réel — Grâce à la personnalisation en temps réel, Amazon Personalize met à jour et adapte les recommandations d'articles en fonction de l'évolution des intérêts de l'utilisateur. Pour de plus amples informations, veuillez consulter Personnalisation en temps réel.
- Exploration — Dans le cadre de l'exploration, les recommandations incluent des éléments comportant moins de données d'interactions ou présentant moins de pertinence pour l'utilisateur. Avec User-Personalization-v 2, Amazon Personalize gère la configuration de l'exploration pour vous. Pour vous assurer que les recommandations incluent de nouveaux articles, vous pouvez utiliser les promotions pour inclure de nouveaux articles en fonction de leur date de création. Pour plus d'informations sur les promotions, consultezPromouvoir des articles dans le cadre de recommandations en temps réel.
- Mises à jour automatiques : grâce aux mises à jour automatiques, Amazon Personalize met automatiquement à jour le dernier modèle (version de la solution) toutes les deux heures afin de prendre en compte les nouveaux articles et de les recommander. Pour de plus amples informations, veuillez consulter Mises à jour automatiques.
- Métadonnées avec recommandations — Avec la recette User-Personalization-v 2, si vous disposez d'un jeu de données d'articles contenant au moins une colonne de métadonnées, les campagnes ont automatiquement la possibilité d'inclure les métadonnées des articles dans les résultats des recommandations. Vous n'avez pas activé manuellement les métadonnées pour votre campagne. Vous pouvez utiliser les métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant les genres de films aux carrousels. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.

Ensembles de données obligatoires et facultatifs

Pour utiliser le User-Personalization-v 2, vous devez créer un jeu de données d'interactions entre articles et importer au moins 1 000 interactions entre articles. Amazon Personalize génère des recommandations principalement basées sur les données d'interaction entre les articles. Pour plus d'informations, consultezDonnées d'interaction entre les articles. User-Personalization-v2 peuvent s'entraîner sur jusqu'à 5 millions d'objets dans le cadre d'interactions entre objets et de jeux de données d'articles.

Avec User-Personalization-v 2, Amazon Personalize peut utiliser les données relatives aux interactions avec les articles, notamment les suivantes :

- Données relatives au type d'événement et à la valeur de l'événement : Amazon Personalize utilise les données relatives aux types d'événements, telles que les types d'événements à cliquer ou à regarder, pour identifier les intentions et les intérêts des utilisateurs en fonction de leurs comportements. Vous pouvez également utiliser les données relatives au type et à la valeur de l'événement pour filtrer les enregistrements avant l'entraînement. Pour de plus amples informations, veuillez consulter Données relatives au type d'événement et à la valeur de l'événement.

  > **Note**
  >
  > Avec User-Personalization-v 2, votre coût de formation est basé sur les données de vos interactions avant d'être filtré par type ou valeur d'événement. Pour plus d'informations sur la tarification, consultez la section Tarification Amazon Personalize.

- Métadonnées contextuelles — Les métadonnées contextuelles sont des données d'interaction que vous collectez sur l'environnement de l'utilisateur au moment d'un événement, telles que sa localisation ou son type d'appareil. Pour de plus amples informations, veuillez consulter Métadonnées contextuelles.

Les ensembles de données suivants sont facultatifs et peuvent améliorer les recommandations :

- Ensemble de données utilisateurs — Amazon Personalize peut utiliser les données de votre jeu de données utilisateurs pour mieux comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Utilisateurs pour filtrer les recommandations. Pour plus d'informations sur les données utilisateur que vous pouvez importer, consultezMétadonnées utilisateur.

- Ensemble de données d'articles : Amazon Personalize peut utiliser les données de votre jeu de données d'articles pour identifier les connexions et les modèles de comportement. Cela permet à Amazon Personalize de comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Items pour filtrer les recommandations. Pour plus d'informations sur les données d'articles que vous pouvez importer, consultezMétadonnées de l'article.

Propriétés et hyperparamètres

La recette User-Personalization-v 2 possède les propriétés suivantes :

- Nom – `aws-user-personalization-v2`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-user-personalization-v2`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-user-personalization-v2`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyperparamètres de la recette User-Personalization-v 2. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Avec User-Personalization-v 2, si vous activez l'entraînement automatique, Amazon Personalize effectue automatiquement le HPO tous les 90 jours. Sans entraînement automatique, aucun HPO ne se produit.

Le tableau fournit les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

Name (Nom)	Description
Hyperparamètres de l'algorithme	

Name (Nom)	Description
`apply_recency_bias`	Détermine si le modèle doit accorder plus de poids aux données d'interactions entre articles les plus récentes de votre jeu de données d'interactions entre articles. Les données d'interactions les plus récentes peuvent inclure des changements soudains dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `apply_recency_bias` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `apply_recency_bias` sur `false`. Valeur par défaut : `true` Plage : `true` ou `false` Type de valeur : booléen HPO personnalisable : non

Recette de personnalisation par l'utilisateur

> **Important**
>
> Nous vous recommandons d'utiliser la recette User-Personalization-v2. Il peut prendre en compte jusqu'à 5 millions d'éléments grâce à une formation plus rapide et générer des recommandations plus pertinentes avec une latence moindre.

La recette User-Personalization (aws-user-personalization) est optimisée pour tous les scénarios de recommandation personnalisés. Il prédit les éléments avec lesquels un utilisateur est le plus susceptible d'interagir. Vous pouvez utiliser la personnalisation utilisateur pour générer des recommandations de films personnalisées pour une application de streaming ou des recommandations de produits personnalisées pour une application de vente au détail.

Avec User-Personalization, Amazon Personalize génère des recommandations principalement basées sur les données d'interaction entre les utilisateurs et les articles dans un ensemble de données d'interactions entre les articles. Il peut également utiliser n'importe quelle métadonnée d'élément et d'utilisateur dans vos ensembles de données Éléments et Utilisateurs. Pour plus d'informations sur les données qu'il utilise, consultezEnsembles de données obligatoires et facultatifs.

Rubriques

- Caractéristiques de la recette
- Ensembles de données obligatoires et facultatifs
- Propriétés et hyperparamètres
- Formation avec la recette de personnalisation utilisateur (console)
- Formation avec la recette de personnalisation utilisateur (SDK Python)
- Obtenir des recommandations et enregistrer des impressions (SDK pour Python (Boto3))
- Exemple de carnet Jupyter

Caractéristiques de la recette

La personnalisation par l'utilisateur utilise les fonctionnalités de recette Amazon Personalize suivantes pour générer des recommandations d'articles :

- Personnalisation en temps réel — Grâce à la personnalisation en temps réel, Amazon Personalize met à jour et adapte les recommandations d'articles en fonction de l'évolution des intérêts de l'utilisateur. Pour de plus amples informations, veuillez consulter Personnalisation en temps réel.
- Exploration — Dans le cadre de l'exploration, les recommandations incluent de nouveaux éléments ou des éléments contenant moins de données d'interactions. Cela améliore la découverte des articles et l'engagement lorsque vous avez un catalogue en évolution rapide, ou lorsque de nouveaux articles, tels que des articles de presse ou des promotions, sont plus pertinents pour les utilisateurs lorsqu'ils sont nouveaux. Pour plus d'informations sur l'exploration, consultezExploration.
- Mises à jour automatiques : grâce aux mises à jour automatiques, Amazon Personalize met automatiquement à jour le dernier modèle (version de la solution) toutes les deux heures afin de prendre en compte les nouveaux articles à recommander. Pour de plus amples informations, veuillez consulter Mises à jour automatiques.

Ensembles de données obligatoires et facultatifs

Pour utiliser la personnalisation utilisateur, vous devez créer un jeu de données d'interactions entre articles et importer au moins 1 000 interactions entre articles. Amazon Personalize génère des recommandations principalement basées sur les données d'interaction entre les articles.

Grâce à la personnalisation par l'utilisateur, Amazon Personalize peut utiliser les données relatives aux interactions avec les articles, notamment les suivantes :

- Données relatives au type et à la valeur de l'événement : Amazon Personalize utilise les données relatives aux types d'événements, telles que les types d'événements à cliquer ou à regarder, pour identifier les intentions et les intérêts des utilisateurs en fonction de leurs comportements. Vous pouvez également utiliser les données relatives au type et à la valeur de l'événement pour filtrer les enregistrements avant l'entraînement. Pour de plus amples informations, veuillez consulter Données relatives au type d'événement et à la valeur de l'événement.
- Métadonnées contextuelles — Les métadonnées contextuelles sont des données d'interaction que vous collectez sur l'environnement de l'utilisateur au moment d'un événement, telles que sa localisation ou son type d'appareil. Pour de plus amples informations, veuillez consulter Métadonnées contextuelles.
- Données relatives aux impressions : les impressions sont des listes d'articles visibles par un utilisateur lorsqu'il interagissait avec un article en particulier (cliqué, regardé, acheté, etc.). Pour de plus amples informations, veuillez consulter Données sur les impressions.

Les ensembles de données suivants sont facultatifs et peuvent améliorer les recommandations :

- Ensemble de données utilisateurs — Amazon Personalize peut utiliser les données de votre jeu de données utilisateurs pour mieux comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Utilisateurs pour filtrer les recommandations. Pour plus d'informations sur les données utilisateur que vous pouvez importer, consultezMétadonnées utilisateur.
- Ensemble de données d'articles : Amazon Personalize peut utiliser les données de votre jeu de données d'articles pour identifier les connexions et les modèles de comportement. Cela permet à Amazon Personalize de comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Items pour filtrer les recommandations. Pour plus d'informations sur les données d'articles que vous pouvez importer, consultezMétadonnées de l'article.

Propriétés et hyperparamètres

La recette de personnalisation utilisateur possède les propriétés suivantes :

- Nom – `aws-user-personalization`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-user-personalization`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-user-personalization`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyperparamètres de la recette de personnalisation utilisateur. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d'organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO tunable (HPO réglable) : le paramètre peut-il participer à HPO ?

Nom	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimensions masquées lorsque votre jeu de données d'interac tions entre objets inclut des modèles plus complexes

Nom	Description
	. L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Pour choisir le meilleur rapport qualité-prix, utilisez HPO. Pour utiliser HPO, définissez `performHPO` sur `true` lorsque vous appelez les opérations CreateSolution et CreateSolutionVersion. Valeur par défaut : 149 Plage : [32, 256] Type de valeur : entier HPO personnalisable : oui

Nom	Description
`bptt`	Détermine s'il convient d'utiliser la technique de propagation de retour à travers le temps. La propagation du retour au fil du temps est une technique qui met à jour les pondérations dans les algorithmes récurrents basés sur les réseaux neuronaux. Utilisez `bptt` pour les crédits à long terme afin de connecter les récompenses retardées à des événements précoces. Par exemple, une récompense retardée peut être un achat effectué après plusieurs clics. Un événement précoce peut être un clic initial. Même dans les mêmes types d'événements, comme un clic, il est judicieux de prendre en compte les effets à long terme et d'optimiser le total des récompenses. Pour prendre en compte les effets à long terme, utilisez des valeurs `bptt` plus élevées. L'utilisation d'une valeur `bptt` plus importante nécessite des ensembles de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 32 Plage : [2, 32] Type de valeur : entier HPO personnalisable : oui

Nom	Description
`recency_mask`	Détermine si le modèle doit prendre en compte les dernières tendances de popularité dans le jeu de données d'interactions entre articles. Les dernières tendances de popularité peuvent inclure des modifications soudaines dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `recency_mask` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `recency_mask` sur `false`. Pour obtenir de bonnes recommandations en utilisant un poids égal, vous pouvez avoir besoin d'un ensemble de données de formation plus grand. Valeur par défaut : `True` Plage : `True` ou `False` Type de valeur : booléen HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Nom	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min_user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Nom	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique, car les données de ces utilisateurs ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min_user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.99 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non
Hyperparamètres de configuration de la campagne d'exploration d'objets	

Nom	Description
`exploration_weight`	Détermine la fréquence à laquelle les recommand ations incluent des éléments présentant moins de données d'interaction ou de pertinence. Plus la valeur est proche de 1,0, plus l'exploration est intense. À zéro, aucune exploration n'a lieu et les recommand ations sont basées sur les données actuelles (pertinen ce). Pour de plus amples informations, veuillez consulter the section called “CampaignConfig”. Valeur par défaut : 0.3 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Nom	Description
`exploration_item_age_cut_of f`	Spécifiez l'âge maximal des objets en jours depuis la dernière interaction entre tous les éléments du jeu de données des interactions entre articles. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge d'un article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquante s, des données d'interaction avec l'article. Pour plus d'informations sur la façon dont Amazon Personali ze détermine l'âge d'un article, consultezDonnées d'horodatage de création. Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommand ations peuvent inclure des articles plus anciens que l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés. Valeur par défaut : 30.0 Plage : valeurs flottantes positives Type de valeur : flottante HPO personnalisable : non

Formation avec la recette de personnalisation utilisateur (console)

Pour utiliser la recette de personnalisation utilisateur afin de générer des recommandations dans la console, commencez par entraîner une nouvelle version de solution à l'aide de cette recette. Déployez ensuite une campagne à l'aide de la version de la solution et utilisez-la pour obtenir des recommandations.

Formation à une nouvelle version de solution à l'aide de la recette de personnalisation utilisateur (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Créez un groupe de jeux de données personnalisé avec un nouveau schéma et téléchargez votre jeu de données avec les données d'impressions. Vous pouvez éventuellement inclure CREATION_TIMESTAMP et les Métadonnées de texte non structurées données dans votre jeu de données Articles afin qu'Amazon Personalize puisse calculer avec plus de précision l'âge d'un article et identifier les articles froids.

   Pour plus d'informations sur l'importation de données, consultezImportation de données de formation dans les ensembles de données Amazon Personalize.
3. Sur la page Groupes de jeux de données, choisissez le nouveau groupe de jeux de données qui contient le ou les ensembles de données contenant des données d'impressions.
4. Dans le volet de navigation, sélectionnez Solutions et recettes, puis sélectionnez Créer une solution.
5. Sur la page Créer une solution, pour le nom de la solution, entrez le nom de votre nouvelle solution.
6. Dans Type de solution, choisissez Recommandation d'article pour obtenir des recommandations d'articles pour vos utilisateurs.
7. Pour Recipe, choisissez aws-user-personalization. La section Configuration de la solution apparaît et propose plusieurs options de configuration.
8. Dans Configuration des événements, si votre ensemble de données d'interactions avec les articles contient les colonnes EVENT_TYPE ou EVENT_TYPE et EVENT_VALUE, utilisez éventuellement les champs Type d'événement et Seuil de valeur d'événement pour choisir les données d'interactions entre articles utilisées par Amazon Personalize lors de la formation du modèle. Pour de plus amples informations, veuillez consulter Choix des données d'interaction entre les objets utilisées pour l'entraînement.

   Si vous avez plusieurs types d'événements et que vous utilisez la recette User-Personalization-v 2 ou Personalized-Ranking-v 2 recettes, vous pouvez également spécifier des pondérations différentes pour les différents types. Par exemple, vous pouvez configurer une solution pour donner plus de poids aux événements d'achat qu'aux événements de clics. Pour de plus amples informations, veuillez consulter Optimisation d'une solution avec configuration d'événements.

9. Configurez éventuellement des hyperparamètres pour votre solution. Pour obtenir la liste des propriétés et des hyperparamètres des recettes de personnalisation utilisateur, consultez. Propriétés et hyperparamètres
10. Choisissez Create and train solution pour commencer la formation. La page Tableau de bord s'affiche.

    Vous pouvez accéder à la page des détails de la solution pour suivre la progression de la formation dans la section Versions de la solution. Lorsque l'entraînement est terminé, le statut est Actif.

Création d'une campagne et obtention de recommandations (console)

Lorsque le statut de la version de votre solution est Actif, vous êtes prêt à créer votre campagne et à obtenir les recommandations suivantes :

1. Sur la page de détails de la solution ou sur la page Campagnes, choisissez Créer une nouvelle campagne.
2. Sur la page Créer une nouvelle campagne, pour les détails de la campagne, fournissez les informations suivantes :

   - Nom de la campagne : entrez le nom de la campagne. Le texte que vous saisissez ici apparaît sur le tableau de bord et la page de détails de la campagne.
   - Solution : Choisissez la solution que vous venez de créer.
   - ID de version de solution : Choisissez l'ID de la version de solution que vous venez de créer.
   - Minimum de transactions provisionnées par seconde : définissez le nombre minimum de transactions provisionnées par seconde pris en charge par Amazon Personalize. Pour plus d'informations, consultez l'CreateCampaignopération.
3. Pour la configuration de la campagne, fournissez les informations suivantes :

   - Poids d'exploration : configurez la quantité à explorer, les recommandations incluant les éléments présentant moins de données d'interaction ou de pertinence, d'autant plus fréquemment que vous spécifiez le niveau d'exploration. Plus la valeur est proche de 1, plus l'exploration est intense. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
   - Âge limite de l'objet d'exploration : entrez l'âge maximum de l'objet, en jours depuis la dernière interaction, pour définir l'étendue de l'exploration de l'objet. Pour augmenter le nombre

d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure.

Par exemple, si vous entrez 10, seuls les éléments dont les données d'interaction entre les éléments datent des 10 jours écoulés depuis la dernière interaction dans le jeu de données sont pris en compte lors de l'exploration.

> **Note**
>
> Les recommandations peuvent inclure des éléments sans données d'interaction entre les éléments en dehors de cette période. Cela est dû au fait que ces éléments correspondent aux intérêts de l'utilisateur et qu'il n'était pas nécessaire de les explorer pour les identifier.

4. Choisissez Create campaign (Créer une campagne).
5. Sur la page des détails de la campagne, lorsque le statut de la campagne est Actif, vous pouvez utiliser la campagne pour obtenir des recommandations et enregistrer des impressions. Pour plus d'informations, consultez Étape 5 : Obtenir des recommandations la section « Mise en route ».

   Amazon Personalize met automatiquement à jour la dernière version de votre solution toutes les deux heures pour inclure les nouvelles données. Votre campagne utilise automatiquement la version mise à jour de la solution. Pour de plus amples informations, veuillez consulter Mises à jour automatiques.

   Pour mettre à jour manuellement la campagne, vous devez d'abord créer et entraîner une nouvelle version de la solution à l'aide de la console ou de l'CreateSolutionVersionopération, avec `trainingMode` set to`update`. Vous mettez ensuite manuellement à jour la campagne sur la page Campagne de la console ou en utilisant l'UpdateCampaignopération.

   > **Note**
   >
   > Amazon Personalize ne met pas automatiquement à jour les versions des solutions que vous avez créées avant le 17 novembre 2020.

Formation avec la recette de personnalisation utilisateur (SDK Python)

Une fois que vous avez créé un groupe de jeux de données et que vous avez chargé vos jeux de données contenant des données d'impressions, vous pouvez élaborer une solution à l'aide de la recette de personnalisation utilisateur. Vous pouvez éventuellement inclure CREATION_TIMESTAMP et les Métadonnées de texte non structurées données dans votre jeu de données Articles afin qu'Amazon Personalize puisse calculer avec plus de précision l'âge d'un article et identifier les articles froids. Pour plus d'informations sur la création de groupes de jeux de données et le téléchargement de données d'entraînement, consultezCréation de fichiers JSON pour les schémas Amazon Personalize.

Pour élaborer une solution à l'aide de la recette de personnalisation utilisateur à l'aide du SDK AWS

1. Créez une nouvelle solution à l'aide de `create_solution` cette méthode.

   `solution name`Remplacez-le par le nom de votre solution et `dataset group arn` par le nom de ressource Amazon (ARN) de votre groupe de données.

   ```
   import boto3

   personalize = boto3.client('personalize')

   print('Creating solution')
   create_solution_response = personalize.create_solution(name = 'solution name',
                              recipeArn = 'arn:aws:personalize:::recipe/aws-user-
   personalization',
                              datasetGroupArn = 'dataset group arn',
                              )
   solution_arn = create_solution_response['solutionArn']
   print('solution_arn: ', solution_arn)
   ```

   Pour obtenir la liste des propriétés et des hyperparamètres des aws-user-personalization recettes, consultezPropriétés et hyperparamètres.

2. Créez une nouvelle version de solution avec les données d'entraînement mises à jour et configurez `trainingMode` pour `FULL` utiliser l'extrait de code suivant. Remplacez le `solution arn` par l'ARN de votre solution.

   ```
   import boto3

   personalize = boto3.client('personalize')
   ```

```
create_solution_version_response = personalize.create_solution_version(solutionArn
 = 'solution arn',
                                                        trainingMode='FULL')

new_solution_version_arn = create_solution_version_response['solutionVersionArn']
print('solution_version_arn:', new_solution_version_arn)
```

3. Lorsque Amazon Personalize a fini de créer la version de votre solution, créez votre campagne avec les paramètres suivants :

   - Fournissez un nouveau `campaign name` et le `solution version arn` généré à l'étape 2.
   - Modifiez l'hyperparamètre de configuration de l'exploration des `explorationWeight` objets pour configurer la quantité à explorer. Les éléments présentant moins de données d'interaction ou de pertinence sont recommandés plus fréquemment lorsque la valeur est proche de 1,0. La valeur par défaut est 0,3.
   - Modifiez le paramètre hyperparamètre de configuration de l'exploration des `explorationItemAgeCutOff` objets pour indiquer la durée maximale, en jours par rapport à la dernière interaction, pendant laquelle les éléments doivent être explorés. Plus la valeur est élevée, plus le nombre d'objets pris en compte lors de l'exploration est important.

   Utilisez l'extrait de code Python suivant pour créer une nouvelle campagne mettant l'accent sur l'exploration, avec une limite d'exploration de 30 jours. La création d'une campagne prend généralement quelques minutes, mais peut prendre plus d'une heure.

```
import boto3

personalize = boto3.client('personalize')

create_campaign_response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn',
    minProvisionedTPS = 1,
    campaignConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
 "explorationItemAgeCutOff": "30"}}
)

campaign_arn = create_campaign_response['campaignArn']
print('campaign_arn:', campaign_arn)
```

Grâce à la personnalisation utilisateur, Amazon Personalize met automatiquement à jour la version de votre solution toutes les deux heures pour inclure les nouvelles données. Votre campagne utilise automatiquement la version mise à jour de la solution. Pour de plus amples informations, veuillez consulter Mises à jour automatiques.

Pour mettre à jour manuellement la campagne, vous devez d'abord créer et entraîner une nouvelle version de la solution à l'aide de la console ou de l'CreateSolutionVersionopération, avec `trainingMode` set to`update`. Vous mettez ensuite manuellement à jour la campagne sur la page Campagne de la console ou en utilisant l'UpdateCampaignopération.

> **Note**
>
> Amazon Personalize ne met pas automatiquement à jour les versions des solutions que vous avez créées avant le 17 novembre 2020.

Obtenir des recommandations et enregistrer des impressions (SDK pour Python (Boto3))

Lorsque votre campagne est créée, vous pouvez l'utiliser pour obtenir des recommandations pour un utilisateur et enregistrer ses impressions. Pour plus d'informations sur l'obtention de recommandations par lots à l'aide de la AWS SDKs sectionCréation d'une tâche d'inférence par lots ()AWS SDKs.

Pour obtenir des recommandations et enregistrer les impressions

1. Appelez la méthode `get_recommendations`. Remplacez `campaign arn` l'ARN de votre nouvelle campagne par `user id` l'userID de l'utilisateur.

```
import boto3

rec_response = personalize_runtime.get_recommendations(campaignArn = 'campaign
 arn', userId = 'user id')
print(rec_response['recommendationId'])
```

2. Créez un nouveau système de suivi des événements pour envoyer PutEvents des demandes. `event tracker name`Remplacez-le par le nom de votre outil de suivi d'événements et `dataset group arn` par l'ARN de votre groupe de données.

```
import boto3

personalize = boto3.client('personalize')

event_tracker_response = personalize.create_event_tracker(
    name = 'event tracker name',
    datasetGroupArn = 'dataset group arn'
)
event_tracker_arn = event_tracker_response['eventTrackerArn']
event_tracking_id = event_tracker_response['trackingId']
print('eventTrackerArn:{},\n eventTrackingId:{}'.format(event_tracker_arn,
 event_tracking_id))
```

3. Utilisez `recommendationId` les étapes 1 et 2 pour créer une nouvelle `PutEvents` demande. `event tracking id` Cette demande enregistre les nouvelles données d'impression issues de la session de l'utilisateur. Remplacez `user id` le par l'ID de l'utilisateur.

```
import boto3

personalize_events.put_events(
     trackingId = 'event tracking id',
     userId= 'user id',
     sessionId = '1',
     eventList = [{
     'sentAt': datetime.now().timestamp(),
     'eventType' : 'click',
     'itemId' : rec_response['itemList'][0]['itemId'],
     'recommendationId': rec_response['recommendationId'],
     'impression': [item['itemId'] for item in rec_response['itemList']],
     }]
)
```

Exemple de carnet Jupyter

Pour un exemple de bloc-notes Jupyter qui montre comment utiliser la recette de personnalisation utilisateur, voir Personnalisation utilisateur avec exploration.

Recette Trending-Now

La recette Trending-Now (aws-trending-now) génère des recommandations pour les articles qui gagnent rapidement en popularité auprès de vos utilisateurs. Vous pouvez utiliser la recette Trending-Now si les articles qui gagnent en popularité sont plus pertinents pour vos clients. Par exemple, vos clients peuvent accorder une grande importance à ce avec quoi les autres utilisateurs interagissent. Les utilisations courantes incluent la recommandation de contenu viral sur les réseaux sociaux, d'articles de dernière minute ou de vidéos sportives récentes.

Trending-Now identifie automatiquement les articles les plus populaires en calculant l'augmentation des interactions entre chaque élément sur des intervalles de temps configurables. Les articles présentant le taux d'augmentation le plus élevé sont considérés comme des articles tendance. L'heure est basée sur les données d'horodatage de votre jeu de données d'interactions entre les articles. Les éléments pris en compte proviennent des données d'interactions que vous avez importées en masse et de manière incrémentielle. Il n'est pas nécessaire de créer manuellement une nouvelle version de solution pour Trending-Now pour prendre en compte de nouveaux éléments dans les données d'interactions.

Vous pouvez spécifier l'intervalle de temps en indiquant le `Trend discovery frequency` moment où vous créez votre solution. Par exemple, si vous spécifiez `30 minutes` pour`Trend discovery frequency`, toutes les 30 minutes de données, Amazon Personalize identifie les articles présentant le taux d'augmentation des interactions le plus élevé depuis la dernière évaluation. Les fréquences possibles incluent 30 minutes, 1 heure, 3 heures et 1 jour. Choisissez une fréquence qui correspond à la distribution de vos données d'interactions. Les données manquantes pendant l'intervalle que vous avez choisi peuvent réduire la précision des recommandations. Si vous n'importez aucune interaction au cours des deux derniers intervalles de temps, Amazon Personalize recommande uniquement les articles populaires plutôt que les articles tendance.

Avec Trending-Now, vous pouvez appeler l'GetRecommendationsopération ou obtenir des recommandations sur la page de campagne de test de la console Amazon Personalize. Amazon Personalize renvoie les articles les plus populaires. Vous ne transmettez une `userId` requête que si vous appliquez un filtre qui l'exige. Avec l' GetRecommendations API, vous pouvez configurer le nombre d'éléments tendance renvoyés avec le `numResults` paramètre. Vous ne pouvez pas obtenir de recommandations par lots avec la recette Trending-Now.

Pour utiliser Trending-Now, vous devez créer un jeu de données d'interactions entre éléments contenant au moins 1 000 interactions historiques et événementielles uniques combinées (après avoir filtré par EventType et eventValueThreshold, si fourni). Lors de la génération de

recommandations d'articles tendance, Trending-Now n'utilise pas les données des ensembles de données Articles ou Utilisateurs. Toutefois, vous pouvez toujours filtrer les recommandations en fonction des données de ces ensembles de données. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Rubriques

- Propriétés et hyperparamètres
- Création d'une solution (SDK pour Python (Boto3))
- Exemple de carnet Jupyter

Propriétés et hyperparamètres

La recette Trending-Now possède les propriétés suivantes :

- Nom – `aws-trending-now`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-trending-now`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-trending-now-custom`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyperparamètres de la recette Trending-Now. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO tunable (HPO réglable) : le paramètre peut-il participer à HPO ?

Nom	Description
Hyperparamètres de transformation des fonctionnalités	
`Trend discovery frequency`	Spécifiez la fréquence à laquelle Amazon Personalize évalue vos données d'interactions et identifie les articles tendance. Par exemple, si vous spécifiez `30 minutes` pour`Trend discovery frequency` , toutes les 30 minutes, Amazon Personalize identifie les articles présentant le taux d'augmentation le plus élevé du nombre d'interactions sur des intervalles de 30 minutes. Les fréquences disponibles incluent 30 minutes, 1 heure, 2 heures, 3 heures et 1 jour. Choisissez une fréquence qui correspond à la distribution de vos données d'interactions. Les données manquantes pendant l'intervalle que vous avez choisi peuvent réduire la précision des recommandations. Si vous utilisez l'opération CreateSolution API sans spécifier de valeur, la valeur par défaut est toutes les 2 heures. **Note** Pour toute valeur de paramètre supérieure à 2 heures, Amazon Personalize actualise automatiquement les recommandations d'articles tendance toutes les 2 heures pour tenir compte des nouvelles interactions et des nouveaux articles. Valeur par défaut : 2 heures Valeurs possibles : 30 minutes, 1 heure, 2 heures, 3 heures et 1 jour. Type de valeu : chaîne

Nom	Description
	HPO personnalisable : non

Création d'une solution (SDK pour Python (Boto3))

Le code suivant montre comment créer une solution avec la recette Trending-Now à l'aide du SDK pour Python (Boto3). Les valeurs possibles pour `trend_discovery_frequency` sont `30 minutes1 hour`,`3 hours`, et`1 day`. Pour plus d'informations sur la création d'une solution à l'aide de la console, consultezCréation d'une solution (console).

```
import boto3

personalize = boto3.client("personalize")

create_solution_response = personalize_client.create_solution(
    name="solution name",
    recipeArn="arn:aws:personalize:::recipe/aws-trending-now",
    datasetGroupArn="dataset group ARN",
    solutionConfig={
        "featureTransformationParameters": {
            "trend_discovery_frequency": "1 hour"
         }
    }
)
print(create_solution_response['solutionArn'])
```

Exemple de carnet Jupyter

Pour un exemple de bloc-notes Jupyter expliquant comment utiliser la recette Trending-Now, consultez trending_now_example.ipynb dans le référentiel d'échantillons Amazon Personalize. GitHub

Recette Popularity-Count

Popularity-Count recommande les articles les plus populaires en fonction de vos données d'interactions. Les éléments les plus populaires sont ceux qui contiennent le plus de données d'interactions provenant d'utilisateurs uniques. La recette renvoie les mêmes éléments populaires pour tous les utilisateurs. Popularity-Count constitue une bonne base de comparaison avec d'autres recettes à l'aide des indicateurs d'évaluation générés par Amazon Personalize lorsque vous créez

une version de solution. Pour de plus amples informations, veuillez consulter Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques.

Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec Popularity-Count, vous devez créer manuellement une nouvelle version de solution (redéfinir le modèle) pour Amazon Personalize afin de prendre en compte les nouveaux articles à recommander et de mettre à jour le modèle en fonction du comportement le plus récent de votre utilisateur. Vous devez ensuite mettre à jour toute campagne à l'aide de la version de la solution. Pour de plus amples informations, veuillez consulter Maintien de la pertinence des recommandations.

Cette recette prédéfinie a les propriétés suivantes :

- Nom – `aws-popularity-count`
- Recette ARN — `arn:aws:personalize:::recipe/aws-popularity-count`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-popularity-count`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/sims`
- Type de recette — `USER_PERSONALIZATION`

Popularity-Count n'a aucun hyperparamètre exposé.

Recette de classement V2 personnalisée

Les Personalized-Ranking-v 2 recettes génèrent des classements personnalisés des articles. Un classement personnalisé est une liste d'éléments recommandés qui sont reclassés en fonction de leur pertinence pour un utilisateur spécifique. Cela est utile si vous disposez d'une collection d'articles commandés, tels que des résultats de recherche, des promotions ou des listes sélectionnées, et que vous souhaitez proposer un reclassement personnalisé pour chacun de vos utilisateurs.

Personalized-Ranking-v2 peuvent s'entraîner sur jusqu'à 5 millions d'objets à partir des interactions entre objets et des ensembles de données d'objets. Et il génère des classements plus précis avec une latence inférieure àPersonalized-Ranking.

Lorsque vous utilisez Personalized-Ranking-v 2, vous spécifiez les éléments à classer dans une opération d'GetPersonalizedRankingAPI. Si vous spécifiez des articles sans données d'interaction, Amazon Personalize les renverra sans score de recommandation dans la réponse de l' GetPersonalizedRanking API.

Cette recette utilise une architecture basée sur un transformateur pour entraîner un modèle qui apprend le contexte et suit les relations et les modèles dans vos données. Les transformateurs sont un type d'architecture de réseau neuronal qui transforme ou modifie une séquence d'entrée en séquence de sortie. Pour Amazon Personalize, la séquence de saisie est l'historique des interactions d'un utilisateur avec les articles dans vos données. La séquence de sortie correspond à leurs recommandations personnalisées. Pour plus d'informations sur les transformateurs, voir Que sont les transformateurs dans l'intelligence artificielle ? dans le AWS Cloud Computing Concepts Hub.

Personalized-ranking-V2 utilise un modèle de tarification différent de celui des autres recettes. Pour plus d'informations sur la tarification, consultez la section Tarification Amazon Personalize.

Rubriques

- Caractéristiques de la recette
- Ensembles de données obligatoires et facultatifs
- Propriétés et hyperparamètres

Caractéristiques de la recette

Personalized-ranking-V2 utilise les fonctionnalités de recette Amazon Personalize suivantes pour classer les articles :

- Personnalisation en temps réel — Grâce à la personnalisation en temps réel, Amazon Personalize met à jour et adapte les recommandations d'articles en fonction de l'évolution des intérêts de l'utilisateur. Pour de plus amples informations, veuillez consulter Personnalisation en temps réel.
- Métadonnées avec recommandations — Avec la recette Personalized-Ranking-v 2, si vous disposez d'un jeu de données d'articles contenant au moins une colonne de métadonnées, les campagnes ont automatiquement la possibilité d'inclure les métadonnées des articles dans les résultats des recommandations. Vous n'avez pas activé manuellement les métadonnées pour votre campagne. Vous pouvez utiliser les métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant les genres de films aux carrousels. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.

Ensembles de données obligatoires et facultatifs

Pour utiliser le Personalized-Ranking-v 2, vous devez créer un jeu de données sur les interactions entre les articles et importer au moins 1 000 interactions entre les articles. Amazon Personalize

génère des classements principalement sur la base des données d'interaction entre les articles. Pour plus d'informations, consultezDonnées d'interaction entre les articles. Personalized-Ranking-v2 peuvent s'entraîner sur jusqu'à 5 millions d'objets dans le cadre d'interactions entre objets et de jeux de données d'articles.

Avec Personalized-Ranking-v 2, Amazon Personalize peut utiliser les données relatives aux interactions avec les articles, notamment les suivantes :

- Données relatives au type d'événement et à la valeur de l'événement : Amazon Personalize utilise les données relatives aux types d'événements, telles que les types d'événements à cliquer ou à regarder, pour identifier les intentions et les intérêts des utilisateurs en fonction de leurs comportements. Vous pouvez également utiliser les données relatives au type et à la valeur de l'événement pour filtrer les enregistrements avant l'entraînement. Pour de plus amples informations, veuillez consulter Données relatives au type d'événement et à la valeur de l'événement.

  > **Note**
  >
  > Avec Personalized-Ranking-v 2, votre coût de formation est basé sur les données de vos interactions avant d'être filtré par type ou valeur d'événement. Pour plus d'informations sur la tarification, consultez la section Tarification Amazon Personalize.

- Métadonnées contextuelles — Les métadonnées contextuelles sont des données d'interaction que vous collectez sur l'environnement de l'utilisateur au moment d'un événement, telles que sa localisation ou son type d'appareil. Pour de plus amples informations, veuillez consulter Métadonnées contextuelles.

Les ensembles de données suivants sont facultatifs et peuvent améliorer les recommandations :

- Ensemble de données utilisateurs — Amazon Personalize peut utiliser les données de votre jeu de données utilisateurs pour mieux comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Utilisateurs pour filtrer les recommandations. Pour plus d'informations sur les données utilisateur que vous pouvez importer, consultezMétadonnées utilisateur.
- Ensemble de données d'articles : Amazon Personalize peut utiliser les données de votre jeu de données d'articles pour identifier les connexions et les modèles de comportement. Cela permet à Amazon Personalize de comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un jeu de données Items pour filtrer les recommandations. Pour

plus d'informations sur les données d'articles que vous pouvez importer, consultezMétadonnées de l'article.

Propriétés et hyperparamètres

La recette Personalized-Ranking-v 2 possède les propriétés suivantes :

- Nom – `aws-personalized-ranking-v2`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-personalized-ranking-v2`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-personalized-ranking-v2`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyperparamètres de la recette Personalized-Ranking-v 2. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Avec Personalized-Ranking-v 2, si vous activez l'entraînement automatique, Amazon Personalize effectue automatiquement le HPO tous les 90 jours. Sans entraînement automatique, aucun HPO ne se produit.

Le tableau fournit les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`apply_recency_bias`	Détermine si le modèle doit accorder plus de poids aux données d'interactions entre articles les plus récentes de votre jeu de données d'interactions entre articles. Les données d'interactions les plus récentes peuvent

Name (Nom)	Description
	inclure des changements soudains dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `apply_recency_bias` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `apply_recency_bias` sur `false`. Valeur par défaut : `true` Plage : `true` ou `false` Type de valeur : booléen HPO personnalisable : non

Recette de classement personnalisé

> **Important**
>
> Nous vous recommandons d'utiliser la recette Personalized-ranking-V2. Il peut prendre en compte jusqu'à 5 millions d'objets grâce à un entraînement plus rapide et générer des classements plus précis avec une latence moindre.

La recette du classement personnalisé génère des classements personnalisés des objets. Un classement personnalisé est une liste d'éléments recommandés ayant fait l'objet d'un nouveau classement pour un utilisateur spécifique. Cela est utile si vous disposez d'une collection d'articles commandés, tels que des résultats de recherche, des promotions ou des listes sélectionnées, et que vous souhaitez proposer un reclassement personnalisé pour chacun de vos utilisateurs. Par exemple, avec Personalized-Ranking, Amazon Personalize peut reclasser les résultats de recherche que vous avez générés. OpenSearch

Pour entraîner un modèle, la recette Personalized-Ranking utilise les données de votre jeu de données d'interactions entre articles, et si vous les avez créées, le jeu de données Items et le jeu de

données Users de votre groupe de jeux de données (ces ensembles de données sont facultatifs). Avec Personalized-Ranking, votre jeu de données d'objets peut inclure Métadonnées de texte non structurées et votre jeu de données d'interactions entre objets peut inclure. Métadonnées contextuelles Pour obtenir un classement personnalisé, utilisez l'GetPersonalizedRankingAPI.

Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec Personalized-Ranking, vous devez créer manuellement une nouvelle version de solution (modifier le modèle) pour qu'Amazon Personalize prenne en compte les nouveaux articles à des fins de recommandation et mette à jour le modèle en fonction du comportement le plus récent de votre utilisateur. Vous devez ensuite mettre à jour toute campagne à l'aide de la version de la solution. Pour de plus amples informations, veuillez consulter Maintien de la pertinence des recommandations.

> **Note**
>
> Si vous fournissez des articles sans données d'interaction à des fins de classement, Amazon Personalize renverra ces articles sans note de recommandation dans la réponse de l' GetPersonalizedRanking API.

Cette recette possède les propriétés suivantes :

- Nom – `aws-personalized-ranking`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-personalized-ranking`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-personalized-ranking`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Type de recette — `PERSONALIZED_RANKING`

Hyperparamètres

Le tableau suivant décrit les hyperparamètres de la recette Personalize Ranking. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d'organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des

hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO personnalisable : le paramètre peut-il participer à l'optimisation des hyperparamètres (HPO) ?

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimensions masquées lorsque votre jeu de données d'interac tions entre objets inclut des modèles plus complexes . L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Pour décider de la valeur optimale, utilisez HPO. Pour utiliser HPO, définisse z `performHPO` sur `true` lorsque vous appelez les opérations CreateSolution et CreateSolutionVersion. Valeur par défaut : 149 Plage : [32, 256] Type de valeur : entier HPO personnalisable : oui
`bptt`	Détermine s'il convient d'utiliser la technique de propagation de retour à travers le temps. La propagati

Name (Nom)	Description
	on du retour au fil du temps est une technique qui met à jour les pondérations dans les algorithmes récurrent s basés sur les réseaux neuronaux. Utilisez `bptt` pour les crédits à long terme afin de connecter les récompenses retardées à des événements précoces. Par exemple, une récompense retardée peut être un achat effectué après plusieurs clics. Un événement précoce peut être un clic initial. Même dans les mêmes types d'événements, comme un clic, il est judicieux de prendre en compte les effets à long terme et d'optimis er le total des récompenses. Pour prendre en compte les effets à long terme, utilisez des valeurs `bptt` plus élevées. L'utilisation d'une valeur `bptt` plus importante nécessite des ensembles de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 32 Plage : [2, 32] Type de valeur : entier HPO personnalisable : oui

Name (Nom)	Description
`recency_mask`	Détermine si le modèle doit prendre en compte les dernières tendances de popularité dans le jeu de données d'interactions entre articles. Les dernières tendances de popularité peuvent inclure des modifications soudaines dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `recency_mask` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `recency_mask` sur `false`. Pour obtenir de bonnes recommandations en utilisant un poids égal, vous pouvez avoir besoin d'un ensemble de données de formation plus grand. Valeur par défaut : `True` Plage : `True` ou `False` Type de valeur : booléen HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Name (Nom)	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Name (Nom)	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique, car les données de ces utilisateurs ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.99 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Exemple de carnet de notes à classement personnalisé

Pour un exemple de bloc-notes Jupyter expliquant comment utiliser la recette de classement personnalisé, voir Exemple de personnalisation du classement.

Recette de similarité sémantique

La recette Semantic-Similarity (aws-semantic-similarity) génère des recommandations pour les éléments sémantiquement similaires à un élément donné en fonction du contenu textuel. Contrairement aux recettes de similarité traditionnelles qui reposent sur les interactions entre l'utilisateur et les éléments, cette recette analyse les descriptions textuelles et les attributs des éléments pour générer des intégrations et identifier les éléments sémantiquement similaires.

Cette recette est idéale pour les scénarios dans lesquels vous souhaitez recommander des articles en fonction de la similitude du contenu, par exemple en recommandant des livres ayant des thèmes similaires, des articles sur des sujets connexes ou des produits avec des descriptions similaires. Cela fonctionne particulièrement bien pour les nouveaux éléments dont l'historique des interactions est limité (scénarios de démarrage à froid) et pour les catalogues où les relations sémantiques sont plus importantes que les modèles de cooccurrence.

Avec Semantic-Similarity, vous fournissez un identifiant d'article lors d'une GetRecommendationsopération (ou dans la console Amazon Personalize) et Amazon Personalize renvoie une liste d'articles similaires. Vous pouvez également utiliser un flux de travail par lots pour obtenir des articles similaires pour tous les articles de votre inventaire (voir Obtenir des recommandations d'articles par lots).

Rubriques

- Caractéristiques de la recette
- Ensembles de données obligatoires et facultatifs
- Propriétés et hyperparamètres

Caractéristiques de la recette

Semantic-Similarity utilise les fonctionnalités de recette Amazon Personalize suivantes pour générer des recommandations d'articles :

- Personnalisation en temps réel — Grâce à la recette Semantic-Similarity, Amazon Personalize met automatiquement à jour votre catalogue d'articles. Lorsque vous ajoutez de nouveaux éléments à votre jeu de données d'articles ou que vous mettez à jour les métadonnées d'éléments existantes, ces modifications sont reflétées dans vos recommandations dans un délai d'environ 30 minutes lorsque vous utilisez la formation incrémentielle. Cela garantit que vos clients voient toujours les articles les plus récents disponibles dans votre catalogue sans avoir à intervenir manuellement ou

à attendre un cycle de recyclage complet. Cela est particulièrement utile pour les catalogues qui changent fréquemment, tels que les articles de presse, les articles de blog ou les offres de produits saisonnières. Pour activer les mises à jour incrémentielles, les clients doivent :

- `performIncrementalUpdate`Défini sur `true` pour la solution dans l'API
- Choisissez l'option « Entraînement complet et progressif » ou « Entraînement progressif » sous Méthode d'entraînement dans l'interface utilisateur

Notez que l'activation des mises à jour incrémentielles entraîne des coûts supplémentaires chaque fois qu'une mise à jour est effectuée.

- Métadonnées avec recommandations : avec la recette de similarité sémantique, les campagnes ont automatiquement la possibilité d'inclure les métadonnées des articles dans les résultats des recommandations. Vous n'avez pas activé manuellement les métadonnées pour votre campagne. Vous pouvez utiliser les métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant les genres de films aux carrousels. Pour plus d'informations, consultez la section Métadonnées des éléments dans les recommandations.

Ensembles de données obligatoires et facultatifs

Pour utiliser la recette Semantic-Similarity, vous devez créer un jeu de données Items. Amazon Personalize génère des recommandations basées sur la signification sémantique des métadonnées des articles. Pour plus d'informations, consultez la section Métadonnées de l'élément. Semantic-Similarity peut s'entraîner sur un maximum de 10 millions d'éléments dans votre jeu de données d'articles.

Avec Semantic-Similarity, Amazon Personalize a besoin de données sur les articles qui incluent les éléments suivants

- Champ ItemName — Un champ de chaîne `itemName` défini sur. `true` Ce champ doit contenir le titre ou le nom de l'article.
- Champ de description textuelle — Au moins un champ de chaîne marqué comme `textual` contenant la description de l'élément. Il doit s'agir du champ qui décrit et représente le mieux l'article.

Amazon Personalize utilise ce champ pour générer des intégrations sémantiques qui capturent le sens et le contenu de vos articles.

En outre, le champ réservé CREATION_TIMESTAMP doit être défini si vous souhaitez utiliser un classement basé sur la fraîcheur. Pour de plus amples informations, veuillez consulter Propriétés et hyperparamètres.

Les ensembles de données suivants sont facultatifs et peuvent améliorer les recommandations :

- Ensemble de données d'interactions : Amazon Personalize peut utiliser les données de votre jeu de données Interactions pour calculer les scores de popularité en fonction de l'engagement des utilisateurs à l'égard des articles. Vous pouvez utiliser les scores de popularité pour classer des articles similaires en fonction de leur popularité auprès des utilisateurs. Vous devez fournir un jeu de données Interactions si vous souhaitez utiliser le classement basé sur la popularité. Vous pouvez également utiliser les données d'un jeu de données Interactions pour filtrer les recommandations. Pour plus d'informations sur les données d'interaction que vous pouvez importer, consultez la section Données d'interaction entre les articles

Propriétés et hyperparamètres

La recette Semantic-Similarity possède les propriétés suivantes :

- Nom – `aws-semantic-similarity`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-semantic-similarity`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-semantic-similarity`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/aws-semantic-similarity`
- Type de recette — `RELATED_ITEMS`

Pour de plus amples informations, veuillez consulter Choisir une recette.

La recette Semantic-Similarity ne présente aucun hyperparamètre, mais vous pouvez configurer des facteurs de popularité et de fraîcheur lorsque vous créez une campagne afin d'influencer le classement d'articles similaires.

Le tableau fournit les informations suivantes pour chaque facteur :

- Plage: [lower bound, upper bound]

- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

Nom	Description
Actualité	Le facteur de fraîcheur représente la récente date d'un article. La fraîcheur est calculée en normalisa nt l'âge de l'article en fonction de son CREATION_ TIMESTAMP. Pour utiliser le facteur de fraîcheur, vous devez inclure le champ CREATION_TIMESTAMP dans le schéma de votre jeu de données Items. Des valeurs plus élevées du facteur de fraîcheur donneront la priorité aux nouveaux articles parmi les recommand ations sémantiquement similaires Valeur par défaut : `0.0` Gamme : `[0.0, 1.0]` Type de valeur : Double
Popularité	Le facteur de popularité représente la popularité d'un article en fonction des interactions des utilisateurs. La popularité est calculée en normalisant le nombre d'interactions reçues par chaque élément. Pour utiliser le facteur de popularité, vous devez inclure un jeu de données Interactions lors de la création de votre groupe de jeux de données. Les valeurs les plus élevées du facteur de popularité donnent la priorité aux articles faisant l'objet d'un plus grand nombre d'interac tions avec les clients parmi des recommandations sémantiquement similaires. Valeur par défaut : `0.0` Gamme : `[0.0, 1.0]` Type de valeur : Double

Notez que les scores de fraîcheur et de popularité sont calculés lors de l'entraînement et que les mises à jour progressives ne mettent pas à jour les scores de popularité et de fraîcheur. Pour que les facteurs de popularité et de fraîcheur les plus récents influencent le classement des articles recommandés, utilisez le recyclage automatique ou redéfinissez manuellement la solution et mettez à jour la campagne avec la nouvelle version de la solution.

Recette d'articles similaires

> **Note**
>
> Toutes les recettes RELATED_ITEMS utilisent des données d'interactions. Choisissez Similar-Items si vous avez également des métadonnées relatives à des articles et que vous souhaitez qu'Amazon Personalize les utilise pour rechercher des articles similaires. Vous pouvez également choisir le Recette SIMS si vous souhaitez configurer d'autres hyperparamètres pour le modèle.

La recette Similar-Items (aws-similar-items) génère des recommandations pour les articles similaires à un article que vous spécifiez. Utilisez Similar-Items pour aider les clients à découvrir de nouveaux articles dans votre catalogue en fonction de leur comportement antérieur et des métadonnées des articles. La recommandation d'articles similaires peut augmenter l'engagement des utilisateurs, le taux de clics et le taux de conversion de votre application.

Similar-Items calcule la similarité en fonction des données d'interactions et de toutes les métadonnées d'article que vous fournissez. Il prend en compte la cooccurrence de l'élément dans l'historique des utilisateurs de votre jeu de données Interaction, ainsi que toute similitude entre les métadonnées de l'élément. Par exemple, avec Similar-Items, Amazon Personalize peut recommander des articles que les clients achètent fréquemment avec un style similaire (Métadonnées catégoriques), ou des films que différents utilisateurs ont également regardés avec une description similaire ()Métadonnées de texte non structurées.

Avec Similar-Items, vous fournissez un identifiant d'article lors d'une GetRecommendations opération (ou dans la console Amazon Personalize) et Amazon Personalize renvoie une liste d'articles similaires. Vous pouvez également utiliser un flux de travail par lots pour obtenir des articles similaires pour tous les articles de votre inventaire (voirObtenir des recommandations d'articles par lots). Lorsque vous obtenez des articles similaires, vous pouvez les filtrer en fonction d'un attribut de l'article que vous spécifiez dans votre demande. Pour ce faire, ajoutez un`CurrentItem`. `attribute`élément de votre filtre. Pour obtenir un exemple, consultez item data filter examples.

Pour utiliser Similar-Items, vous devez créer un jeu de données d'interactions entre articles contenant au moins 1 000 interactions historiques et événementielles uniques (combinées). Pour des prévisions plus précises, nous vous recommandons également de créer un jeu de données d'articles et d'importer des métadonnées relatives aux articles de votre catalogue. Similar-Items n'utilise pas les données d'un ensemble de données Users lors de la génération de recommandations. Vous pouvez toujours filtrer les recommandations en fonction des données d'un ensemble de données Utilisateurs. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Si vous disposez d'un jeu de données d'articles contenant des données textuelles et des données de titre d'article, vous pouvez générer des thèmes pour les articles connexes dans des recommandations par lots. Pour de plus amples informations, consultez Recommandations par lots avec des thèmes issus du générateur de contenu.

Vous pouvez obtenir des recommandations pour des objets similaires à un objet froid (un objet avec moins de cinq interactions). Si Amazon Personalize ne trouve pas le numéro d'article que vous avez indiqué dans votre demande de recommandation ou dans votre fichier de saisie par lots, la recette renvoie les articles populaires sous forme de recommandations.

Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec Similar-Items, vous devez créer manuellement une nouvelle version de solution (modifier le modèle) pour Amazon Personalize afin de prendre en compte les nouveaux articles à des fins de recommandation et de mettre à jour le modèle en fonction du comportement le plus récent de votre utilisateur. Vous devez ensuite mettre à jour toute campagne à l'aide de la version de la solution. Pour de plus amples informations, veuillez consulter Maintien de la pertinence des recommandations.

Propriétés et hyperparamètres

La recette Similar-Items possède les propriétés suivantes :

- Nom – `aws-similar-items`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-similar-items`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-similar-items`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyperparamètres de la recette Similar-Items. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO tunable (HPO réglable) : le paramètre peut-il participer à HPO ?

Nom	Description
Hyperparamètres de l'algorithme	
`popularity_discount_factor`	Configurez l'influence de la popularité sur les recommandations. Spécifiez une valeur proche de zéro pour inclure les articles les plus populaires. Spécifiez une valeur plus proche de 1 pour mettre moins l'accent sur la popularité. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non
`item_id_hidden_dim`	Le nombre de variables masquées qu'Amazon Personalize utilise pour modéliser les identifiants d'articles intégrés en fonction des données d'interac tions. Les variables masquées recréent l'historique des achats des utilisateurs et les statistiques des articles pour générer des scores de classement. Pour l'utilise `ritem_id_hidden_dim` , vous devez utiliser HPO

Nom	Description
	et fournir des valeurs de plage minimales et maximales . Amazon Personalize utilise HPO pour trouver la meilleure valeur dans la plage que vous spécifiez. Spécifiez une valeur maximale supérieure lorsque vous disposez d'un ensemble de données d'interac tions entre articles volumineux. L'utilisation d'une valeur maximale supérieure nécessite plus de temps de traitement. Pour utiliser HPO, réglez-le sur `performHPO` le `true` moment où vous appelez l'CreateSolutionopération. Valeur par défaut : 100 Gamme : [30, 200] Type de valeur : entier HPO personnalisable : oui

Nom	Description
`item_metadata_hidden_dim`	Le nombre de variables masquées qu'Amazon Personalize utilise pour modéliser les métadonnées des articles. Pour l'utiliser`item_metadata_hidden_dim` , vous devez utiliser HPO et fournir des valeurs de plage minimales et maximales. Amazon Personalize utilise HPO pour trouver la meilleure valeur dans la plage que vous spécifiez. Spécifiez une valeur maximale supérieure lorsque vous disposez d'un ensemble de données d'interactions entre articles volumineux. L'utilisation d'un maximum supérieur nécessite plus de temps de traitement. Pour utiliser HPO, réglez-le sur `performHPO` le `true` moment où vous appelez l'CreateSolutionopération. Valeur par défaut : 100 Gamme : [30, 200] Type de valeur : entier HPO personnalisable : oui

Recette SIMS

> **Note**
>
> Toutes les recettes RELATED_ITEMS utilisent des données d'interactions. Choisissez SIMS si vous souhaitez configurer davantage d'hyperparamètres pour le modèle. Choisissez le Recette d'articles similaires si vous avez des métadonnées d'article et que vous souhaitez qu'Amazon Personalize les utilise pour rechercher des articles similaires.

La recette de Item-to-item similarités (SIMS) utilise un filtrage collaboratif pour recommander les éléments les plus similaires à un élément que vous spécifiez lorsque vous recevez des recommandations. SIMS utilise votre jeu de données sur les interactions entre les articles, et non les

métadonnées des articles telles que la couleur ou le prix, pour déterminer la similitude. SIMS identifie la cooccurrence de l'élément dans l'historique des utilisateurs de votre jeu de données Interaction afin de recommander des éléments similaires. Par exemple, avec les cartes SIM, Amazon Personalize peut recommander des articles de café que les clients achètent fréquemment ensemble ou des films que différents utilisateurs regardent également.

Lorsque vous recevez des recommandations d'articles similaires, vous pouvez filtrer les articles en fonction d'un attribut de l'article que vous spécifiez dans votre demande. Pour ce faire, ajoutez un`CurrentItem`. `attribute`élément de votre filtre. Pour obtenir un exemple, consultez item data filter examples.

Pour utiliser SIMS, vous devez créer un jeu de données d'interactions entre éléments contenant au moins 1 000 interactions historiques et événementielles uniques (combinées). SIMS n'utilise pas les données d'un jeu de données Users or Items lors de la génération de recommandations. Vous pouvez toujours filtrer les recommandations en fonction des données contenues dans ces ensembles de données. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Si les données relatives au comportement des utilisateurs sont insuffisantes pour un article ou si le numéro d'article que vous avez fourni est introuvable, SIMS recommande des articles populaires. Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec SIMS, vous devez créer manuellement une nouvelle version de solution (modifier le modèle) pour Amazon Personalize afin de prendre en compte les nouveaux éléments à recommander et de mettre à jour le modèle en fonction du comportement le plus récent de votre utilisateur. Vous devez ensuite mettre à jour toute campagne à l'aide de la version de la solution. Pour de plus amples informations, veuillez consulter Maintien de la pertinence des recommandations.

La recette SIMS possède les propriétés suivantes :

- Nom – `aws-sims`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-sims`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-sims`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/sims`
- Type de recette — `RELATED_ITEMS`

Le tableau suivant décrit les hyper-paramètres de la recette SIMS. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d'organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO personnalisable : le paramètre peut-il participer à l'optimisation des hyperparamètres (HPO) ?

Nom	Description
Hyperparamètres de l'algorithme	
`popularity_discount_factor`	Configurez l'influence de la popularité sur les recommandations. Spécifiez une valeur proche de zéro pour inclure les articles les plus populaires. Spécifiez une valeur plus proche de 1 pour mettre moins l'accent sur la popularité. Valeur par défaut : 0.5 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : oui
`min_cointeraction_count`	Nombre minimal de co-interactions dont vous avez besoin pour calculer la similarité entre une paire d'éléments. Par exemple, la valeur 3 signifie que vous avez besoin de trois utilisateurs ou plus ayant interagi

Nom	Description
	avec les deux éléments pour que l'algorithme calcule leur similarité. Valeur par défaut : 3 Plage : [0, 10] Type de valeur : entier HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Nom	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données disponibles sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Valeur par défaut : 0.005 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Nom	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d’historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données disponibles sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique. Les utilisateurs dont l'historique est long ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, `min_hist_length_percentile = 0.05` et `max_hist_length_percentile = 0.95` incluent tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.995 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Nom	Description
`min_item_interaction_count_percentile`	Le percentile minimal du nombre d'interactions d'éléments à inclure dans la formation du modèle. Utilisez `min_item_interaction_count_percentile` pour exclure un pourcentage d'éléments avec un historique court des interactions. Les éléments avec un historique court sont souvent de nouveaux éléments. Leur suppression peut entraîner des modèles en mettant davantage l'accent sur les éléments ayant un historique connu. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des éléments, mais supprime les cas périphériques. Valeur par défaut : 0.01 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Nom	Description
`max_item_interaction_count_percentile`	Le percentile maximal du nombre d'interactions d'éléments à inclure dans la formation du modèle. Utilisez `max_item_interaction_count_percentile` pour exclure un pourcentage d'éléments avec un historique court des interactions. Les éléments dont l'historique est long ont tendance à être plus anciens et peuvent ne pas être à jour. Par exemple, une version de film qui est épuisée. La suppression de ces éléments peut se concentrer sur des éléments plus pertinents. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des éléments, mais supprime les cas périphériques. Par exemple, `min_item_interaction_count_percentile = 0.05` et `max_item_interaction_count_percentile = 0.95` incluent tous les éléments, à l'exception de ceux dont le nombre d'interactions est inférieur ou supérieur à 5 %. Valeur par défaut : 0.9 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Exemple de bloc-notes SIMS

Pour obtenir un exemple de bloc-notes Jupyter qui vous montre comment utiliser la recette SIMS, veuillez consulter Finding similar items + HPO.

Next-Best-Action recette

La recette Next-Best-Action (aws-next-best-action) génère des recommandations en temps réel pour les prochaines meilleures actions pour vos utilisateurs. La meilleure action suivante pour un utilisateur est celle qu'il est le plus susceptible d'entreprendre. Par exemple, vous inscrire à votre programme de fidélité, télécharger votre application ou demander une carte de crédit.

Vous pouvez ainsi fournir des recommandations d'action personnalisées à vos utilisateurs lorsqu'ils utilisent votre application. Next-Best-Action Suggérer la bonne action à un utilisateur peut inciter un plus grand nombre d'utilisateurs à effectuer vos actions. En fonction des actions que vous souhaitez recommander, vous pouvez fidéliser vos clients, générer plus de revenus et améliorer l'expérience utilisateur de votre application. Pour un exemple de cas d'utilisation décrivant comment les recommandations d'action personnalisées peuvent bénéficier à une application de commerce électronique, voirExemple de cas d'utilisation.

Amazon Personalize prédit la meilleure action suivante à partir des actions que vous importez dans votre jeu de données Actions. Il identifie les actions qu'un utilisateur est le plus susceptible d'entreprendre en fonction de ses interactions avec les actions et les éléments. Si vos données d'action incluent la valeur de l'action, Amazon Personalize prend en compte la valeur de l'action. Si un utilisateur est également susceptible d'effectuer deux actions différentes, Amazon Personalize classe l'action dont la valeur est la plus élevée.

Lorsque vous recevez des recommandations d'actions en temps réel pour un utilisateur, Amazon Personalize renvoie une liste des actions que l'utilisateur est le plus susceptible d'effectuer au cours d'une période configurable (`leaction optimization period`). Par exemple, les mesures qu'ils prendront probablement au cours des 14 prochains jours. La liste est triée par ordre décroissant par indice de propension. Ce score représente la probabilité que l'utilisateur entreprenne l'action.

Jusqu'à ce que vous importiez les données d'interaction entre les actions, Amazon Personalize recommande des actions dans votre boîte sans personnalisation, et les scores de propension sont de 0,0. Une action aura un score une fois qu'elle aura obtenu les résultats suivants :

- Au moins 50 interactions d'action avec le type d'événement TAKEN.
- Au moins 50 interactions d'action avec le type d'événement NOT_TAKEN ou VIEWED.

Ces interactions d'action doivent être présentes lors de la dernière formation sur la version de la solution et doivent avoir lieu dans un délai de 6 semaines à compter de l'horodatage de la dernière interaction dans le jeu de données d'interactions d'action.

Pour plus d'informations sur les données utilisées par la Next-Best-Action recette, consultezEnsembles de données obligatoires et facultatifs.

Lorsque vous créez une solution à l'aide de la Next-Best-Action recette, vous pouvez configurer la fenêtre de temps qu'Amazon Personalize utilise pour prédire les actions à l'aide de l'hyperparamètre de `action optimization period` fonctionnalité. Pour de plus amples informations, veuillez consulter Propriétés et hyperparamètres.

Rubriques

- Exemple de cas d'utilisation
- Caractéristiques de la recette
- Ensembles de données obligatoires et facultatifs
- Propriétés et hyperparamètres

Exemple de cas d'utilisation

Suggérer la bonne action à un utilisateur peut inciter un plus grand nombre d'utilisateurs à effectuer vos actions. En fonction des actions que vous souhaitez recommander, vous pouvez potentiellement fidéliser vos clients, générer plus de revenus et améliorer l'expérience utilisateur de votre application.

Par exemple, vous pouvez avoir une application de commerce électronique qui suggère les différentes actions suivantes :

- Abonnez-vous au programme de fidélité
- Téléchargez l'application mobile
- Achat dans la catégorie Bijoux
- Achat dans la catégorie Beauté et soins

Vous avez peut-être un utilisateur qui fait fréquemment des achats sur votre site et qui a effectué à plusieurs reprises des achats de bijoux, de produits de beauté et de produits de beauté. Pour cet utilisateur, les recommandations d'action d'Amazon Personalize et leurs scores peuvent inclure les éléments suivants :

- Abonnez-vous au programme de fidélité

  Note de propension — 1,00
- Achat dans la catégorie Bijoux

Score de propension — 0,86

- Achat dans la catégorie Beauté et soins

  Score de propension — 0,85

Grâce à ces recommandations d'action, vous savez comment inciter l'utilisateur à s'inscrire à votre programme de fidélité. Cette action a le score de propension le plus élevé et c'est l'action que l'utilisateur entreprendra le plus probablement. Cela est dû au fait que l'utilisateur fait fréquemment des achats dans votre boutique et qu'il est susceptible de profiter des avantages de votre programme de fidélité.

Caractéristiques de la recette

La Next-Best-Action recette utilise les fonctionnalités de recette Amazon Personalize suivantes pour générer des recommandations d'action :

- Personnalisation en temps réel : Amazon Personalize utilise la personnalisation en temps réel pour mettre à jour et adapter les recommandations d'action en fonction de l'évolution des intérêts de l'utilisateur. Pour de plus amples informations, veuillez consulter Personnalisation en temps réel.
- Exploration : avec l'exploration, les recommandations incluent de nouvelles actions ou des actions avec moins de données d'interactions. Pour plus d'informations sur l'exploration, consultezExploration.
- Mises à jour automatiques : grâce aux mises à jour automatiques, Amazon Personalize met automatiquement à jour le dernier modèle (version de la solution) toutes les deux heures afin d'inclure de nouvelles actions dans les recommandations issues de l'exploration. Pour de plus amples informations, veuillez consulter Mises à jour automatiques.

Ensembles de données obligatoires et facultatifs

Pour utiliser la Next-Best-Action recette, vous devez créer les ensembles de données suivants :

- Actions : vous importez des données relatives à vos actions, telles que leur valeur, dans un ensemble de données Amazon Personalize Actions.

  Dans les données de vos actions, vous pouvez fournir un EXPIRATION_TIMESTAMP pour chaque action. Si une action a expiré, Amazon Personalize ne l'inclura pas dans les recommandations. Vous pouvez également fournir une valeur REPEAT_FREQUENCY pour chaque action. Cela

indique le temps qu'Amazon Personalize doit attendre avant de recommander à nouveau une action après qu'un utilisateur interagisse avec celle-ci. Pour plus d'informations sur les données qu'un jeu de données Actions peut stocker, consultezMétadonnées relatives aux actions.

- Interactions entre objets : votre ensemble de données d'interactions entre objets doit comporter au moins 1 000 interactions entre objets. Amazon Personalize utilise les interactions entre les articles pour comprendre l'état actuel de vos utilisateurs et leurs centres d'intérêt. Pour plus d'informations sur les données relatives aux interactions entre les éléments, consultezDonnées d'interaction entre les articles.

Les ensembles de données suivants sont facultatifs :

- Ensemble de données d'interactions d'action : une interaction d'action est une interaction impliquant un utilisateur et une action dans votre jeu de données Actions. Vous pouvez importer les interactions entre les actions entreprises, non entreprises et vues. Bien que ces données soient facultatives, nous vous recommandons d'importer les données d'interaction entre les actions pour obtenir des recommandations de qualité. Si vous ne disposez pas de données d'interaction d'action, vous pouvez créer un ensemble de données d'interactions d'action vide et enregistrer les interactions de vos clients avec les actions à l'aide de l'opération d'PutActionInteractionsAPI.

  Jusqu'à ce que vous importiez les données d'interaction entre les actions, Amazon Personalize recommande des actions dans votre boîte sans personnalisation, et les scores de propension sont de 0,0. Une action aura un score une fois qu'elle aura obtenu les résultats suivants :

  - Au moins 50 interactions d'action avec le type d'événement TAKEN.
  - Au moins 50 interactions d'action avec le type d'événement NOT_TAKEN ou VIEWED.

  Ces interactions d'action doivent être présentes lors de la dernière formation sur la version de la solution et doivent avoir lieu dans un délai de 6 semaines à compter de l'horodatage de la dernière interaction dans le jeu de données d'interactions d'action.

  Pour plus d'informations sur les données d'interactions entre actions que vous pouvez importer, consultezDonnées d'interaction entre actions. Pour plus d'informations sur l'enregistrement des événements d'interaction entre actions, consultezEnregistrement des événements d'interaction en temps réel.

> **Note**
>
> Avec Next-Best-Action, Amazon Personalize n'utilise pas les données d'impressions ni les métadonnées contextuelles dans un ensemble de données d'interactions Action.

- Utilisateurs : Amazon Personalize utilise toutes les données de votre ensemble de données utilisateurs pour mieux comprendre vos utilisateurs et leurs centres d'intérêt. Vous pouvez également utiliser les données d'un ensemble de données Utilisateurs pour filtrer les recommandations d'actions. Pour plus d'informations sur les données utilisateur que vous pouvez importer, consultezMétadonnées utilisateur.
- Articles : Amazon Personalize utilise toutes les données de votre jeu de données Articles ainsi que de votre jeu de données d'interactions avec les articles pour identifier les liens et les modèles de leur comportement. Cela permet à Amazon Personalize de comprendre vos utilisateurs et leurs centres d'intérêt. Pour plus d'informations sur les données d'articles que vous pouvez importer, consultezMétadonnées de l'article.

Propriétés et hyperparamètres

La Next-Best-Action recette ne prend pas en charge l'optimisation des hyperparamètres. La Next-Best-Action recette possède les propriétés suivantes :

- Nom – `aws-next-best-action`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-next-best-action`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-next-best-action`

Le tableau suivant décrit les hyperparamètres de fonctionnalité de la aws-next-best-action recette. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d’organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

- HPO réglable : si le paramètre peut participer au HPO

Name (Nom)	Description
Hyperparamètres de fonctionnalité	
`action_optimization_period`	La fenêtre temporelle utilisée par Amazon Personalize pour prévoir les prochaines meilleures actions pour un utilisateur. Par exemple, les actions que l'utilisateur effectuera probablement au cours des 14 prochains jours. Si vous ne disposez pas de beaucoup de données d'interaction entre actions, spécifiez une valeur supérieure. Si vous n'êtes pas sûr de la valeur à spécifier, utilisez la valeur par défaut. Valeur par défaut : 14 Gamme : [7, 28] Type de valeur : entier HPO personnalisable : non

Recette Item-Affinity

La recette Item-Affinity (aws-item-affinity) est une recette USER_SEGMENTATION qui crée un segment utilisateur (groupe d'utilisateurs) pour chaque élément que vous spécifiez. Il s'agit des utilisateurs qu'Amazon Personalize prévoit qu'ils interagiront le plus probablement avec chaque article. Utilisez Item-Affinity pour en savoir plus sur vos utilisateurs et prendre des mesures en fonction de leurs segments d'utilisateurs respectifs.

Par exemple, vous souhaiterez peut-être créer une campagne marketing pour votre application de vente au détail en fonction des préférences des utilisateurs concernant les articles de votre catalogue. Item-Affinity créerait un segment utilisateur pour chaque élément en fonction des données de vos ensembles de données Interactions et Items. Vous pouvez l'utiliser pour promouvoir différents articles auprès de différents segments d'utilisateurs en fonction de la probabilité qu'ils entreprennent

une action (par exemple, cliquer sur un article ou acheter un article). D'autres utilisations peuvent inclure la vente croisée de produits à différents groupes d'utilisateurs ou l'identification de candidats potentiels.

Pour obtenir des segments d'utilisateurs basés sur les éléments, vous créez une solution et une version de solution avec la recette Item-Affinity, puis vous ajoutez une liste d'éléments au format JSON dans un compartiment Amazon S3 et vous créez une tâche de segment par lots. Amazon Personalize envoie un segment utilisateur pour chaque article à votre emplacement de sortie dans Amazon S3. Vos données d'entrée peuvent contenir un maximum de 500 éléments pour lesquels obtenir des segments d'utilisateurs. Pour plus d'informations sur la préparation des données d'entrée pour une tâche de segmentation par lots, consultezPréparation des données d'entrée pour les recommandations par lots.

Pour utiliser Item-Affinity, vous devez disposer d'un jeu de données sur les interactions entre articles. Les ensembles de données des éléments et des utilisateurs sont facultatifs. Vous pouvez obtenir des segments d'utilisateurs avec des tâches de segmentation par lots. Pour de plus amples informations, veuillez consulter Obtenir des segments d'utilisateurs par lots.

Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec Item-Affinity, vous devez créer une nouvelle version de solution pour Amazon Personalize afin de prendre en compte les nouveaux utilisateurs pour les segments d'utilisateurs et de mettre à jour le modèle en fonction du comportement le plus récent de vos utilisateurs. Pour obtenir un segment d'utilisateur pour un élément, celui-ci doit être présent lorsque vous avez créé la version de la solution.

La recette Item-Affinity possède les propriétés suivantes :

- Nom – `aws-item-affinity`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-item-affinity`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-item-affinity`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/item-affinity`
- Type de recette — `USER_SEGMENTATION`

Le tableau suivant décrit les hyperparamètres de la recette Item-Affinity. Un hyperparamètre est un paramètre d'algorithme que vous ajustez pour améliorer les performances du modèle. Les

hyperparamètres d'algorithme contrôlent les performances du modèle. Vous ne pouvez pas utiliser l'optimisation des hyperparamètres (HPO) avec la recette Item-Affinity.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimension s masquées lorsque votre ensemble de données Interactions inclut des modèles plus complexes. L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 149 Plage : [32, 256] Type de valeur : entier

Item-Attribute-Affinity recette

La recette Item-Attribute-Affinity (aws-item-attribute-affinity) est une recette USER_SEGMENTATION qui crée un segment utilisateur (groupe d'utilisateurs) pour chaque attribut d'article que vous spécifiez. Il s'agit des utilisateurs qui, selon Amazon Personalize, sont les plus susceptibles d'interagir avec les articles dotés de l'attribut en question. Item-Attribute-AffinityÀ utiliser pour en savoir plus sur vos utilisateurs et prendre des mesures en fonction de leurs segments d'utilisateurs respectifs.

Par exemple, vous souhaiterez peut-être créer une campagne marketing pour votre application de vente au détail en fonction des préférences des utilisateurs concernant les types de chaussures de votre catalogue. Item-Attribute-Affinitycréerait un segment utilisateur pour chaque type de chaussure basé sur les données de vos ensembles de données Interactions et Items. Vous pouvez l'utiliser pour promouvoir différentes chaussures auprès de différents segments d'utilisateurs en fonction de la probabilité qu'ils entreprennent une action (par exemple, cliquer sur une chaussure ou acheter une chaussure). D'autres utilisations peuvent inclure la promotion de différents genres de films auprès de différents utilisateurs ou l'identification d'un candidat potentiel en fonction du type d'emploi.

Pour obtenir des segments d'utilisateurs basés sur les attributs des articles, vous créez une solution et une version de solution avec la Item-Attribute-Affinity recette, puis vous ajoutez une liste d'attributs d'article au format JSON dans un compartiment Amazon S3 et vous créez une tâche de segment par lots. Amazon Personalize envoie un segment utilisateur pour chaque article à votre emplacement de sortie dans Amazon S3. Vos données d'entrée peuvent comporter un maximum de 10 requêtes, chaque requête portant sur un ou plusieurs attributs d'article. Pour plus d'informations sur la préparation des données d'entrée pour une tâche de segmentation par lots, consultezPréparation des données d'entrée pour les recommandations par lots.

Vous devez disposer d'un jeu de données d'interactions entre les éléments et d'un ensemble de données d'articles pour les utiliser Item-Attribute-Affinity. Votre jeu de données Items doit comporter au moins une colonne qui est une colonne de métadonnées non textuelle et non réservée. Vous pouvez obtenir des segments d'utilisateurs avec des tâches de segmentation par lots. Pour de plus amples informations, veuillez consulter Obtenir des segments d'utilisateurs par lots avec des ressources personnalisées.

Après avoir créé une version de solution, assurez-vous de maintenir la version et les données de votre solution à jour. Avec Item-Attribute-Affinity, vous devez créer une nouvelle version de solution pour Amazon Personalize afin de prendre en compte les nouveaux utilisateurs pour les segments d'utilisateurs et de mettre à jour le modèle en fonction du comportement le plus récent de vos utilisateurs. Pour obtenir un segment d'utilisateur pour un attribut d'article, l'attribut d'article doit être présent lorsque vous avez créé la version de la solution.

La Item-Attribute-Affinity recette possède les propriétés suivantes :

- Nom – `aws-item-attribute-affinity`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-item-attribute-affinity`

- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-item-attribute-affinity`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/item-attribute-affinity`
- Type de recette — `USER_SEGMENTATION`

Le tableau suivant décrit les hyperparamètres de la Item-Attribute-Affinity recette. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Vous ne pouvez pas utiliser l'optimisation des hyperparamètres (HPO) avec la Item-Attribute-Affinity recette.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimension s masquées lorsque votre ensemble de données Interactions inclut des modèles plus complexes. L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 149 Plage : [32, 256] Type de valeur : entier

Recettes HRNN traditionnelles

Les recettes HRNN traditionnelles ne sont plus disponibles. Cette documentation est fournie à titre de référence.

Nous vous recommandons d'utiliser la recette aws-user-personalizaton (personnalisation utilisateur) plutôt que les anciennes recettes HRNN. La personnalisation par l'utilisateur améliore et unifie les fonctionnalités proposées par les recettes HRNN. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

Amazon Personalize peut choisir automatiquement la recette de réseau neuronal récurrent hiérarchique (HRNN) la plus appropriée en fonction de son analyse des données d'entrée. Cette option est appelée AutoML. Pour de plus amples informations, veuillez consulter Utiliser AutoML pour choisir une recette HRNN (API uniquement).

Rubriques

- Utiliser AutoML pour choisir une recette HRNN (API uniquement)
- Recette HRNN (ancienne)
- Recette de métadonnées HRNN (ancienne)
- Recette HRNN-ColdStart (ancienne)

Utiliser AutoML pour choisir une recette HRNN (API uniquement)

Amazon Personalize peut choisir automatiquement la recette de réseau neuronal récurrent hiérarchique (HRNN) la plus appropriée en fonction de son analyse des données d'entrée. Cette option est appelée AutoML. Pour exécuter AutoML, définissez le paramètre `performAutoML` sur `true` lorsque vous appelez l'API CreateSolution API.

Vous pouvez également spécifier la liste des recettes examinées par Amazon Personalize pour déterminer la recette optimale, en fonction d'une métrique que vous spécifiez. Dans ce cas, vous appelez l'`CreateSolution`opération, vous spécifiez `true` le `performAutoML` paramètre, vous omettez le `recipeArn` paramètre et vous incluez le `solutionConfig` paramètre, en spécifiant le `metricName` et dans le `recipeList` cadre de l'`autoMLConfig`objet.

Le tableau suivant présente la manière dont une recette est choisie. Vous devez spécifier `performAutoML` ou `recipeArn`, mais pas les deux. AutoML HRNN est uniquement exécuté à partir des recettes HRNN.

performAutoML	recipeArn	solutionConfig	Résultat
true	omettre	omis	Amazon Personalize choisit la recette
true	omettre	`autoMLConfig` : `metricName` et `recipeList` spécifiés	Amazon Personalize choisit une recette dans la liste qui optimise la métrique
omettre	spécifié	omis	Vous devez spécifier la recette.
omettre	spécifié	spécifié	Spécifiez la recette et remplacez les propriétés de formation par défaut

> **Note**
>
> Quand `performAutoML` a la valeur `true`, tous les paramètres de l'objet `solutionConfig` sont ignorés sauf `autoMLConfig`.

Recette HRNN (ancienne)

> **Note**
>
> Les recettes HRNN traditionnelles ne sont plus disponibles. Cette documentation est fournie à titre de référence.
> Nous vous recommandons d'utiliser la recette aws-user-personalizaton (personnalisation utilisateur) plutôt que les anciennes recettes HRNN. La personnalisation par l'utilisateur améliore et unifie les fonctionnalités proposées par les recettes HRNN. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

La recette du réseau neuronal récurrent hiérarchique (HRNN) Amazon Personalize modélise les modifications du comportement des utilisateurs afin de fournir des recommandations au cours

d'une session. Une session est un ensemble d'interactions utilisateur au sein d'une période donnée dans le but de trouver un élément spécifique pour répondre à un besoin, par exemple. En évaluant davantage les interactions récentes d'un utilisateur, vous pouvez fournir des recommandations plus pertinentes au cours d'une session.

HRNN tient compte de l'intention et des intérêts de l'utilisateur, qui peuvent changer au fil du temps. Les historiques d'utilisateurs ordonnés sont pondérés automatiquement pour créer de meilleures inférences. HRNN utilise un mécanisme de passage pour modéliser les pondérations de remise en tant que fonction pouvant être apprise des éléments et des horodatages.

Amazon Personalize extrait les fonctionnalités de chaque utilisateur à partir de votre ensemble de données. Si vous avez terminé l'intégration des données en temps réel, ces fonctionnalités sont mises à jour en temps réel en fonction de l'activité de l'utilisateur. Pour obtenir une recommandation, vous fournissez uniquement le `USER_ID`. Si vous en fournissez également un`ITEM_ID`, Amazon Personalize l'ignore.

La recette HRNN possède les propriétés suivantes :

- Nom – `aws-hrnn`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-hrnn`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-hrnn`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Type de recette — `USER_PERSONALIZATION`

Le tableau suivant décrit les hyper-paramètres de la recette HRNN. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d’organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]

- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO tunable (HPO réglable) : le paramètre peut-il participer à HPO ?

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'historique des achats des utilisateurs et les statistiques des articles pour générer des scores de classement. Spécifiez un plus grand nombre de dimensions masquées lorsque votre jeu de données d'interactions entre objets inclut des modèles plus complexes. L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Pour décider de la valeur optimale, utilisez HPO. Pour utiliser HPO, définissez `performHPO` sur `true` lorsque vous appelez les opérations CreateSolution et CreateSolutionVersion. Valeur par défaut : 43 Plage : [32, 256] Type de valeur : entier HPO personnalisable : oui
`bptt`	Détermine s'il convient d'utiliser la technique de propagation de retour à travers le temps. La propagation du retour au fil du temps est une technique qui met à jour les pondérations dans les algorithmes récurrents basés sur les réseaux neuronaux. Utilisez `bptt` pour les crédits à long terme afin de connecter les récompenses retardées à des événements précoces. Par exemple, une récompense retardée peut être un

Name (Nom)	Description
	achat effectué après plusieurs clics. Un événement précoce peut être un clic initial. Même dans les mêmes types d'événements, comme un clic, il est judicieux de prendre en compte les effets à long terme et d'optimiser le total des récompenses. Pour prendre en compte les effets à long terme, utilisez des valeurs `bptt` plus élevées. L'utilisation d'une valeur `bptt` plus importante nécessite des ensembles de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 32 Plage : [2, 32] Type de valeur : entier HPO personnalisable : oui

Name (Nom)	Description
`recency_mask`	Détermine si le modèle doit prendre en compte les dernières tendances de popularité dans le jeu de données d'interactions entre articles. Les dernières tendances de popularité peuvent inclure des modifications soudaines dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `recency_mask` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `recency_mask` sur `false`. Pour obtenir de bonnes recommandations en utilisant un poids égal, vous pouvez avoir besoin d'un ensemble de données de formation plus grand. Valeur par défaut : `True` Plage : `True` ou `False` Type de valeur : booléen HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Name (Nom)	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Name (Nom)	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique, car les données de ces utilisateurs ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.99 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Recette de métadonnées HRNN (ancienne)

> **Note**
>
> Les recettes HRNN traditionnelles ne sont plus disponibles. Cette documentation est fournie à titre de référence.
> Nous vous recommandons d'utiliser la recette aws-user-personalizaton (personnalisation utilisateur) plutôt que les anciennes recettes HRNN. La personnalisation par l'utilisateur améliore et unifie les fonctionnalités proposées par les recettes HRNN. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

La recette HRNN-Metadata prédit les éléments avec lesquels un utilisateur interagira. Elle est semblable à la recette HRNN, avec des fonctionnalités supplémentaires dérivées des métadonnées contextuelles, d'utilisateurs et d'éléments (respectivement issues des ensembles de données Interactions, Utilisateurs et Éléments). HRNN-Metadata fournit une meilleure précision que les autres modèles sans métadonnées lorsque des métadonnées de haute qualité sont disponibles. L'utilisation de cette recette peut nécessiter des durées de formation plus longues.

La recette HRNN-Metadata possède les propriétés suivantes :

- Nom – `aws-hrnn-metadata`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-hrnn-metadata`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-hrnn-metadata`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/featurize_metadata`
- Type de recette — `USER_PERSONALIZATION`

Le tableau suivant décrit les hyper-paramètres de la recette HRNN-Metadata. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d'organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO personnalisable : le paramètre peut-il participer à l'optimisation des hyperparamètres (HPO) ?

Name (Nom)	Description
Hyperparamètres d'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimensions masquées lorsque votre jeu de données d'interac tions entre objets inclut des modèles plus complexes . L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Pour décider de la valeur optimale, utilisez HPO. Pour utiliser HPO, définisse z `performHPO` sur `true` lorsque vous appelez les opérations CreateSolution et CreateSolutionVersion. Valeur par défaut : 43 Plage : [32, 256] Type de valeur : entier HPO personnalisable : oui
`bptt`	Détermine s'il convient d'utiliser la technique de propagation de retour à travers le temps. La propagati on du retour au fil du temps est une technique qui met à jour les pondérations dans les algorithmes récurrent s basés sur les réseaux neuronaux. Utilisez `bptt`

Name (Nom)	Description
	pour les crédits à long terme afin de connecter les récompenses retardées à des événements précoces. Par exemple, une récompense retardée peut être un achat effectué après plusieurs clics. Un événement précoce peut être un clic initial. Même dans les mêmes types d'événements, comme un clic, il est judicieux de prendre en compte les effets à long terme et d'optimis er le total des récompenses. Pour prendre en compte les effets à long terme, utilisez des valeurs `bptt` plus élevées. L'utilisation d'une valeur `bptt` plus importante nécessite des ensembles de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 32 Plage : [2, 32] Type de valeur : entier HPO personnalisable : oui

Name (Nom)	Description
`recency_mask`	Détermine si le modèle doit prendre en compte les dernières tendances de popularité dans le jeu de données d'interactions entre articles. Les dernières tendances de popularité peuvent inclure des modifications soudaines dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `recency_mask` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `recency_mask` sur `false`. Pour obtenir de bonnes recommandations en utilisant un poids égal, vous pouvez avoir besoin d'un ensemble de données de formation plus grand. Valeur par défaut : `True` Plage : `True` ou `False` Type de valeur : booléen HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Name (Nom)	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Name (Nom)	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique, car les données de ces utilisateurs ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.99 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Recette HRNN-ColdStart (ancienne)

> **Note**
>
> Les recettes HRNN traditionnelles ne sont plus disponibles. Cette documentation est fournie à titre de référence.
> Nous vous recommandons d'utiliser la recette aws-user-personalizaton (personnalisation utilisateur) plutôt que les anciennes recettes HRNN. La personnalisation par l'utilisateur améliore et unifie les fonctionnalités proposées par les recettes HRNN. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

Utilisez la recette HRNN-ColdStart pour prévoir les éléments avec lesquels un utilisateur va interagir lorsque vous ajoutez fréquemment de nouveaux éléments et interactions et que vous souhaitez obtenir des recommandations pour ces éléments immédiatement. La recette HRNN-Coldstart est similaire à la recette HRNN-Metadata, mais elle vous permet d'obtenir des recommandations pour de nouveaux éléments.

En outre, vous pouvez utiliser la recette HRNN-Coldstart lorsque vous souhaitez exclure de la formation des éléments qui ont une longue liste d'interactions, soit en raison d'une tendance récente de popularité, soit parce que les interactions peuvent être très inhabituelles et introduire du bruit dans la formation. Avec HRNN-Coldstart, vous pouvez filtrer les éléments moins pertinents pour créer un sous-ensemble pour la formation. Le sous-ensemble d'éléments, appelé éléments froids, est constitué d'éléments présentant des événements d'interaction connexes dans le jeu de données d'interactions entre articles. Un élément est considéré comme un élément froid lorsqu'il contient :

- Moins d'interactions qu'un nombre spécifié d'interactions maximales. Vous spécifiez cette valeur dans l'hyperparamètre `cold_start_max_interactions` de la recette.
- Durée relative plus courte que la durée maximale. Vous spécifiez cette valeur dans l'hyperparamètre `cold_start_max_duration` de la recette.

Pour réduire le nombre d'éléments froids, définissez une valeur inférieure pour `cold_start_max_interactions` ou `cold_start_max_duration`. Pour augmenter le nombre d'éléments froids, définissez une valeur supérieure pour `cold_start_max_interactions` ou `cold_start_max_duration`.

HRNN-Coldstart a les limites suivantes pour les éléments froids :

- `Maximum cold start items` : 80.000
- `Minimum cold start items` : 100

Si le nombre d'éléments froids est en dehors de cette plage, les tentatives de création d'une solution échouent.

La recette HRNN-Coldstart possède les propriétés suivantes :

- Nom – `aws-hrnn-coldstart`
- Nom de la ressource Amazon (ARN) de la recette — `arn:aws:personalize:::recipe/aws-hrnn-coldstart`
- Algorithme ARN — `arn:aws:personalize:::algorithm/aws-hrnn-coldstart`
- ARN de transformation des fonctionnalités — `arn:aws:personalize:::feature-transformation/featurize_coldstart`
- Type de recette — `USER_PERSONALIZATION`

Pour de plus amples informations, veuillez consulter Choisir une recette.

Le tableau suivant décrit les hyper-paramètres de la recette HRNN-Coldstart. Un hyperparamètre est un paramètre d'algorithme que vous pouvez régler pour améliorer les performances du modèle. Les hyperparamètres d'algorithme contrôlent les performances du modèle. Les hyperparamètres d'organisation de fonction contrôlent la façon de filtrer les données à utiliser dans la formation. Le processus de sélection de la meilleure valeur pour un hyperparamètre est appelé optimisation des hyperparamètres (HPO). Pour de plus amples informations, veuillez consulter Hyperparamètres et HPO.

Le tableau fournit également les informations suivantes pour chaque hyperparamètre :

- Plage: [lower bound, upper bound]
- Type de valeur : integer (entier), Continuous (float) [continu (flottant)], Categorical (Boolean, list, string) [catégorique (booléen, liste, chaîne)]
- HPO tunable (HPO réglable) : le paramètre peut-il participer à HPO ?

Name (Nom)	Description
Hyperparamètres de l'algorithme	
`hidden_dimension`	Nombre de variables masquées utilisées dans le modèle. Les variables masquées recréent l'histori que des achats des utilisateurs et les statistiques des articles pour générer des scores de classemen t. Spécifiez un plus grand nombre de dimensions masquées lorsque votre jeu de données d'interac tions entre objets inclut des modèles plus complexes . L'utilisation de dimensions masquées nécessite un ensemble de données plus volumineux et plus de temps pour le traitement. Pour décider de la valeur optimale, utilisez HPO. Pour utiliser HPO, définisse z `performHPO` sur `true` lorsque vous appelez les opérations CreateSolution et CreateSolutionVersion. Valeur par défaut : 149 Plage : [32, 256] Type de valeur : entier HPO personnalisable : oui
`bptt`	Détermine s'il convient d'utiliser la technique de propagation de retour à travers le temps. La propagati on du retour au fil du temps est une technique qui met à jour les pondérations dans les algorithmes récurrent s basés sur les réseaux neuronaux. Utilisez `bptt` pour les crédits à long terme afin de connecter les récompenses retardées à des événements précoces. Par exemple, une récompense retardée peut être un achat effectué après plusieurs clics. Un événement précoce peut être un clic initial. Même dans les mêmes types d'événements, comme un clic, il est judicieux de prendre en compte les effets à long terme et d'optimis

Name (Nom)	Description
	er le total des récompenses. Pour prendre en compte les effets à long terme, utilisez des valeurs `bptt` plus élevées. L'utilisation d'une valeur `bptt` plus importante nécessite des ensembles de données plus volumineux et plus de temps pour le traitement. Valeur par défaut : 32 Plage : [2, 32] Type de valeur : entier HPO personnalisable : oui
`recency_mask`	Détermine si le modèle doit prendre en compte les dernières tendances de popularité dans le jeu de données d'interactions entre articles. Les dernières tendances de popularité peuvent inclure des modificat ions soudaines dans les modèles sous-jacents des événements d'interaction. Pour former un modèle qui accorde plus de poids aux événements récents, définissez `recency_mask` sur `true`. Pour former un modèle qui pèse de manière égale toutes les interactions passées, définissez `recency_mask` sur `false`. Pour obtenir de bonnes recommandations en utilisant un poids égal, vous pouvez avoir besoin d'un ensemble de données de formation plus grand. Valeur par défaut : `True` Plage : `True` ou `False` Type de valeur : booléen HPO personnalisable : oui
Hyperparamètres de fonctionnalité	

Name (Nom)	Description
`cold_start_max_interactions`	Nombre maximal d'interactions utilisateur/élément qu'un élément peut avoir pour être considéré comme un élément froid. Valeur par défaut : 15 Plage : entiers positifs Type de valeur : entier HPO personnalisable : non
`cold_start_max_duration`	Durée maximale en jours par rapport au point de départ pour qu'une interaction utilisateur/élément soit considérée comme un élément de démarrage à froid. Pour définir le point de départ de l'interaction utilisateur/élément, définissez l'hyper-paramètre `cold_start_relative_from` . Valeur par défaut : 5.0 Plage : valeurs flottantes positives Type de valeur : flottante HPO personnalisable : non

Name (Nom)	Description
`cold_start_relative_from`	Détermine le point de départ du calcul `cold_start_max_duration` par la recette HRNN-Coldstart. Pour calculer à partir de l'heure actuelle, choisissez `currentTime` . Pour effectuer le calcul à `cold_start_max_duration` partir de l'horodatage du dernier élément du jeu de données d'interactions entre articles, choisissez. `latestItem` Ce paramètre est utile si vous ajoutez fréquemment de nouveaux éléments. Valeur par défaut : `latestItem` Plage : `currentTime` , `latestItem` Type de valeu : chaîne HPO personnalisable : non

Name (Nom)	Description
`min_user_history_length_percentile`	Le percentile minimal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `min_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec des longueurs d'historique courtes. Les utilisateurs avec un court historique affichent souvent des modèles basés sur la popularité des éléments plutôt que sur les besoins ou les souhaits personnels de l'utilisateur. Leur suppression peut former des modèles en mettant davantage l'accent sur les modèles sous-jacents dans vos données. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.0 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Name (Nom)	Description
`max_user_history_length_percentile`	Le percentile maximal des longueurs d'historique d'utilisateur à inclure dans la formation du modèle. La longueur de l'historique correspond à la quantité totale de données sur un utilisateur. Utilisez `max_user_history_length_percentile` pour exclure un pourcentage d'utilisateurs avec de longues longueurs d'historique, car les données de ces utilisateurs ont tendance à contenir du bruit. Par exemple, un robot peut avoir une longue liste d'interactions automatisées. La suppression de ces utilisateurs limite le bruit dans la formation. Choisissez une valeur appropriée après avoir examiné les longueurs de l'historique utilisateur, à l'aide d'un histogramme ou d'un outil similaire. Nous vous recommandons de définir une valeur qui conserve la majorité des utilisateurs, mais supprime les cas périphériques. Par exemple, la définition de `min__user_history_length_percentile to 0.05` et `max_user_history_length_percentile to 0.95` inclut tous les utilisateurs, sauf ceux dont la longueur de l'historique figure dans les 5 % du bas ou du haut. Valeur par défaut : 0.99 Plage : [0.0, 1.0] Type de valeur : flottante HPO personnalisable : non

Préparation des données de formation pour Amazon Personalize

Après avoir choisi un cas d'utilisation ou une recette de domaine et pris note de ses exigences en matière de données, vous êtes prêt à commencer à préparer vos données. Amazon Personalize peut utiliser les types de données suivants :

- Interactions avec les articles — Dans Amazon Personalize, une interaction avec un article est un événement d'interaction positive entre un utilisateur et un article de votre catalogue. Par exemple, un utilisateur qui regarde un film, consulte une annonce ou achète une paire de chaussures.
- Articles : les métadonnées des articles peuvent inclure des informations telles que le prix, le type de SKU, la description ou la disponibilité de chaque article de votre catalogue.
- Utilisateurs — Les métadonnées utilisateur peuvent inclure des informations telles que l'âge, le sexe, l'adhésion au programme de fidélité et les intérêts de chacun de vos utilisateurs.
- Actions — Une action est une activité d'engagement que vous souhaiterez peut-être recommander à vos clients. Les actions peuvent inclure l'installation de votre application mobile, la création d'un profil de membre, l'adhésion à votre programme de fidélité ou l'inscription à des e-mails promotionnels. Pour la Next-Best-Action recette, le jeu de données Actions est requis. Aucune autre recette personnalisée ou cas d'utilisation de domaine n'utilise les données Actions.
- Interactions d'action — Une interaction d'action est un événement d'interaction entre un utilisateur et une action. La Next-Best-Action recette utilise ces données et celles de votre jeu de données Actions pour recommander des actions à vos utilisateurs. Aucune autre recette personnalisée ou cas d'utilisation de domaine n'utilise les données d'interactions d'action.

Amazon Personalize stocke les données dans des ensembles de données, un pour chaque type de données. Chaque jeu de données a des exigences différentes. Lorsque vous importez des données dans un ensemble de données Amazon Personalize, vous pouvez choisir d'importer des enregistrements en bloc, individuellement ou les deux. Les importations en masse impliquent l'importation d'un grand nombre d'enregistrements historiques stockés dans un ou plusieurs fichiers CSV d'un compartiment Amazon S3.

- Si vous ne disposez pas de données en masse, vous pouvez utiliser des opérations d'importation individuelles pour collecter des données et diffuser des événements jusqu'à ce que vous répondiez aux exigences de formation d'Amazon Personalize et aux exigences en matière de données de votre cas d'utilisation ou de votre recette de domaine. Pour plus d'informations sur l'enregistrement

d'événements, consultezEnregistrement des événements en temps réel pour influencer les recommandations. Pour plus d'informations sur l'importation d'enregistrements individuels, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize.

- Si vous n'êtes pas sûr de disposer de suffisamment de données ou si vous avez des questions quant à leur qualité, vous pouvez importer vos données dans un ensemble de données Amazon Personalize et utiliser Amazon Personalize pour les analyser. Pour de plus amples informations, veuillez consulter Analyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

Les sections suivantes présentent les exigences relatives aux données pour chaque type de jeu de données Amazon Personalize et les directives relatives à la préparation de données en masse. Si vous ne disposez pas de données groupées, consultez les sections pour comprendre les données obligatoires et facultatives que vous pouvez importer dans le cadre d'opérations d'importation individuelles. Si vous avez besoin d'aide supplémentaire pour formater vos données, vous pouvez utiliser Amazon SageMaker AI Data Wrangler (Data Wrangler) pour préparer vos données. Pour de plus amples informations, veuillez consulter Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler.

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Rubriques

- Directives relatives au format des données en masse pour tous les types de données
- Préparation des données d'interaction entre les objets pour la formation
- Préparation des métadonnées des éléments pour la formation
- Préparation des métadonnées utilisateur pour la formation
- Préparation des métadonnées d'action pour la formation
- Préparation des données d'interaction entre les actions pour la formation

Directives relatives au format des données en masse pour tous les types de données

Les directives et exigences suivantes peuvent vous aider à vous assurer que vos données en masse sont correctement formatées.

- Vos données d'entrée doivent se trouver dans un fichier CSV (valeurs séparées par des virgules).
- La première ligne de votre fichier CSV doit contenir les en-têtes de colonne. Ne placez pas les en-têtes entre guillemets (« »).
- Les colonnes doivent avoir des noms alphanumériques uniques. Par exemple, vous ne pouvez pas ajouter à la fois un `GENRES_FIELD_1` champ et un `GENRESFIELD1` champ.
- Si vous importez plusieurs fichiers CSV, tous les en-têtes de colonne doivent correspondre à tous les fichiers.
- Assurez-vous de disposer des champs obligatoires pour votre type de jeu de données et assurez-vous que leurs noms correspondent aux exigences d'Amazon Personalize. Par exemple, les données de vos articles peuvent comporter une colonne appelée `ITEM_IDENTIFICATION_NUMBER` avec IDs pour chacun de vos articles. Pour utiliser cette colonne comme champ ITEM_ID, renommez-la en. `ITEM_ID` Si vous utilisez Data Wrangler pour formater vos données, vous pouvez utiliser les colonnes de carte pour la transformation Amazon Personalize Data Wrangler afin de vous assurer que vos colonnes sont correctement nommées.

  Pour plus d'informations sur l'utilisation de Data Wrangler pour préparer vos données, consultez. Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler
- Chaque enregistrement de votre fichier CSV doit se trouver sur une seule ligne.
- Amazon Personalize ne prend pas en charge les types de données complexes tels que les tableaux et les cartes.
- Pour qu'Amazon Personalize utilise des données booléennes lors de l'entraînement ou du filtrage, utilisez des valeurs `"True"` de chaîne `"False"` et/ou des valeurs numériques `1` pour vrai et `0` pour faux.
- Si vous utilisez Data Wrangler pour formater vos données, vous pouvez utiliser le Data Wrangler pour transformer Parse Value as Type pour convertir les types de données.
- `TIMESTAMP`et `CREATION_TIMESTAMP` les données doivent être au format Epoch Time Unix. Pour de plus amples informations, veuillez consulter Données d'horodatage.
- Évitez d'inclure des " caractères ou des caractères spéciaux dans les données d'ID d'élément, d'ID utilisateur et d'ID d'action.

- Si vos données contiennent des caractères non ASCII, votre fichier CSV doit être codé au format UTF-8.
- Assurez-vous de formater les données textuelles comme décrit dansMétadonnées de texte non structurées.

Préparation des données d'interaction entre les objets pour la formation

Une interaction avec un article est un événement d'interaction positive entre un utilisateur et un article de votre catalogue. Par exemple, un utilisateur qui regarde un film, consulte une annonce ou achète une paire de chaussures. Vous importez des données relatives aux interactions de vos utilisateurs avec vos articles dans un jeu de données sur les interactions entre articles. Vous pouvez enregistrer plusieurs types d'événements, tels que les clics, les visionnages ou les achats.

Par exemple, si un utilisateur clique sur un article en particulier puis l'aime, vous pouvez demander à Amazon Personalize d'utiliser ces événements comme données de formation. Pour chaque événement, vous devez enregistrer l'identifiant de l'utilisateur, l'identifiant de l'élément, l'horodatage (au format Unix Time Epoch) et le type d'événement (click and like). Vous ajouterez ensuite les deux événements d'interaction entre éléments à un jeu de données d'interactions entre articles.

Pour tous les cas d'utilisation du domaine et les recettes personnalisées, les données relatives aux interactions entre articles en vrac doivent figurer dans un fichier CSV. Chaque ligne doit représenter une interaction unique entre un utilisateur et un élément. Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Les sections suivantes fournissent plus d'informations sur la façon de préparer les données d'interaction avec les articles pour Amazon Personalize. Pour les directives relatives au format des données en masse pour tous les types de données, consultez les directives relatives au format des données en masse

Rubriques

- Exigences relatives aux données d'interaction entre les articles
- Données d'horodatage
- Données relatives au type d'événement et à la valeur de l'événement

- Métadonnées contextuelles
- Données sur les impressions
- Exemple de données d'interactions

Exigences relatives aux données d'interaction entre les articles

Les sections suivantes répertorient les exigences relatives aux données d'interaction avec les articles pour Amazon Personalize. Pour des quotas supplémentaires, voirAmazon Personalize : points de terminaison et quotas.

Exigences minimales en matière de formation

Pour tous les cas d'utilisation du domaine et les recettes personnalisées, les données relatives aux interactions entre les articles en vrac doivent être les suivantes :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

Pour créer une recommandation ou une solution personnalisée, vous devez au minimum créer un jeu de données sur les interactions entre les articles.

Exigences relatives aux colonnes

Les données relatives aux interactions avec les articles doivent comporter les colonnes suivantes.

- USER_ID — Identifiant unique de l'utilisateur qui a interagi avec l'élément. Chaque événement doit avoir un USER_ID. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- ITEM_ID — Identifiant unique de l'élément avec lequel l'utilisateur a interagi. Chaque événement doit avoir un identifiant d'article. Il doit s'agir `string` d'une longueur maximale de 256 caractères.

- TIMESTAMP — Heure à laquelle l'événement s'est produit (au format Unix Epoch en secondes). Chaque interaction doit avoir un TIMESTAMP. Pour de plus amples informations, veuillez consulter Données d'horodatage.
- EVENT_TYPE — La nature de l'événement d'interaction avec un objet, tel qu'un clic, une montre ou un achat. Pour les recommandeurs de domaines, vous devez disposer d'une colonne de type d'événement et chaque interaction doit avoir un type d'événement. Pour toutes les recettes personnalisées, une colonne EVENT_TYPE est recommandée mais facultative. Si vous l'ajoutez, chaque événement doit avoir un type d'événement. Pour de plus amples informations, veuillez consulter Données relatives au type d'événement et à la valeur de l'événement.

Vous êtes libre d'ajouter des colonnes personnalisées supplémentaires en fonction de votre cas d'utilisation et de vos données. Le nombre maximum de colonnes de métadonnées facultatives est de 5. Ces colonnes peuvent inclure des empty/null valeurs. Nous recommandons que ces colonnes soient remplies à au moins 70 %.

Données d'horodatage

Les données d'horodatage doivent être au format Epoch Time Unix en secondes. Par exemple, l'horodatage Epoch en secondes pour la date du 31 juillet 2020 est 1596238243. Pour convertir les dates en horodatages d'époque Unix, utilisez un convertisseur Epoch, un convertisseur d'horodatage Unix.

Amazon Personalize utilise les données d'horodatage pour calculer la récence et identifier tout modèle basé sur le temps. Cela permet à Amazon Personalize de suivre les recommandations en fonction up-to-date de l'évolution des préférences des utilisateurs.

Données relatives au type d'événement et à la valeur de l'événement

Un jeu de données d'interactions entre articles peut stocker des données sur le type d'événement et la valeur de l'événement pour chaque interaction. Seules les ressources personnalisées utilisent les données de valeur des événements.

Données relatives au type d'événement

Le type d'événement d'une interaction avec un élément fournit un contexte sur sa nature et son importance. Les exemples de type d'événement peuvent être un clic, un visionnage ou un achat. Amazon Personalize utilise les données relatives aux types d'événements, telles que les données

relatives aux clics ou aux achats, pour identifier les intentions et les intérêts des utilisateurs. Le nombre maximum de types d'événements distincts combiné au nombre total de colonnes de métadonnées facultatives dans un jeu de données d'interactions entre éléments est de 10.

Pour les recommandeurs de domaines, vous devez disposer d'une colonne de type d'événement et chaque interaction doit avoir un type d'événement. Pour toutes les recettes personnalisées, une colonne EVENT_TYPE est recommandée mais facultative. Si vous l'ajoutez, chaque événement doit avoir un type d'événement.

Si vous créez des ressources personnalisées, vous pouvez choisir les événements utilisés pour l'entraînement par type d'événement. Si votre ensemble de données contient plusieurs types d'événements dans une colonne EVENT_TYPE et que vous ne fournissez aucun type d'événement lorsque vous configurez une solution personnalisée, Amazon Personalize utilise toutes les données relatives aux interactions entre les articles pour l'entraînement avec le même poids, quel que soit le type. Pour de plus amples informations, veuillez consulter Choix des données d'interaction entre les objets utilisées pour l'entraînement.

Si vous avez plusieurs types d'événements et que vous utilisez la recette User-Personalization-v 2 ou Personalized-Ranking-v 2 recettes, lorsque vous configurez une solution personnalisée, vous pouvez spécifier des pondérations différentes pour les différents types. Par exemple, vous pouvez configurer une solution pour donner plus de poids aux événements d'achat qu'aux événements de clics. Pour de plus amples informations, veuillez consulter Optimisation d'une solution avec configuration d'événements.

Les cas d'utilisation suivants ont des exigences spécifiques en matière de type d'événement :

Cas d'utilisation du domaine VIDEO_ON_DEMAND

- Parce que vous avez regardé X, il faut au moins 1 000 `Watch` événements.
- Le plus populaire nécessite au moins 1 000 `Watch` événements.

Cas d'utilisation de domaines de commerce électronique

- La plupart des visionnages nécessitent au moins 1 000 `View` événements.
- Les meilleures ventes nécessitent au moins 1 000 `Purchase` événements.

Types d'événements positifs et négatifs

Amazon Personalize part du principe que toute interaction est positive. Les interactions avec un type d'événement négatif, tel que l'aversion, n'empêcheront pas nécessairement l'élément d'apparaître dans les futures recommandations de l'utilisateur.

Voici comment les événements négatifs et le désintérêt des utilisateurs peuvent influencer les recommandations :

- Pour tous les cas d'utilisation du domaine et pour la Personnalisation par l'utilisateur recette, Amazon Personalize peut utiliser les données d'impressions. Lorsqu'un élément apparaît dans les données d'impressions et qu'un utilisateur ne le choisit pas, il est moins susceptible d'apparaître dans les recommandations. Pour de plus amples informations, veuillez consulter Données sur les impressions.
- Si vous utilisez des ressources personnalisées et que vous importez des types d'événements positifs et négatifs, vous pouvez vous entraîner uniquement sur les types d'événements positifs, puis filtrer les éléments avec lesquels l'utilisateur a interagi négativement. Pour plus d'informations, consultez Choix des données d'interaction entre les objets utilisées pour l'entraînement et Recommandations de filtrage et segments d'utilisateurs.

Données relatives à la valeur des événements (ressources personnalisées)

Les données relatives à la valeur de l'événement peuvent être le pourcentage d'un film regardé par un utilisateur ou une note sur 10. Si vous créez des solutions personnalisées, vous pouvez choisir les enregistrements utilisés pour l'entraînement en fonction des données des colonnes EVENT_TYPE et EVENT_VALUE. Avec les recommandations de domaines, Amazon Personalize n'utilise pas les données relatives à la valeur des événements et vous ne pouvez pas filtrer les événements avant l'entraînement.

Pour choisir des enregistrements en fonction du type et de la valeur, enregistrez les données relatives au type d'événement et à la valeur de l'événement. Tous les événements ne doivent pas nécessairement avoir une valeur d'événement. La valeur que vous choisissez pour chaque événement dépend des données que vous souhaitez exclure et des types d'événements que vous enregistrez. Par exemple, vous pouvez faire correspondre l'activité de l'utilisateur, par exemple le pourcentage de vidéos regardées par l'utilisateur pour les types d'événements à regarder.

Lorsque vous configurez une solution, vous définissez une valeur spécifique comme seuil pour exclure les enregistrements de la formation. Par exemple, si vos données EVENT_VALUE pour

les événements ayant un EVENT_TYPE de visionnage correspondent au pourcentage d'une vidéo regardée par un utilisateur, si vous définissez le seuil de valeur de l'événement à 0,5 et le type d'événement à regarder, Amazon Personalize entraîne le modèle en utilisant uniquement les événements d'interaction avec le watch dont la valeur EVENT_VALUE est supérieure ou égale à 0,5.

Pour de plus amples informations, consultez Choix des données d'interaction entre les objets utilisées pour l'entraînement.

Métadonnées contextuelles

Avec certaines recettes et certains cas d'utilisation des recommandations, Amazon Personalize peut utiliser des métadonnées contextuelles pour identifier les modèles sous-jacents qui révèlent les éléments les plus pertinents pour vos utilisateurs. Les métadonnées contextuelles sont des données d'interactions que vous collectez sur l'environnement de l'utilisateur au moment d'un événement, telles que sa localisation ou son type d'appareil. Vous pouvez également spécifier le contexte d'un utilisateur lorsque vous recevez des recommandations pour celui-ci.

Incluez des métadonnées contextuelles pour offrir une expérience plus personnalisée à vos utilisateurs et réduire la phase de démarrage à froid pour les nouveaux utilisateurs. La phase de démarrage à froid est celle où les recommandations sont moins pertinentes en raison du manque de données historiques sur les utilisateurs.

Par exemple, si le fichier CSV de vos interactions avec les articles inclut une colonne DEVICE_TYPE contenant des phone valeurs `tablet` et, Amazon Personalize peut découvrir comment les clients font leurs achats différemment selon les appareils. Lorsque vous recevez des recommandations pour un utilisateur, vous pouvez spécifier son appareil et les recommandations seront plus pertinentes, même si l'utilisateur n'a aucun historique d'interactions.

Ce qui suit montre comment formater un fichier CSV d'interactions entre articles avec une colonne DEVICE_TYPE en tant que métadonnées contextuelles.

```
ITEM_ID,USER_ID,TIMESTAMP,DEVICE_TYPE,EVENT_TYPE
shoe12345,12,1428624000,Tablet,CLICK
shoe12346,12,1420416000,Tablet,CLICK
shoe12347,12,1410652800,Tablet,BUY
shoe4444,13,1409961600,Phone,CLICK
shoe4445,13,1402876800,Phone,BUY
shoe4336,13,1402185600,Phone,CLICK
.....
```

Pour les groupes de jeux de données de domaine, les cas d'utilisation des recommandations suivants peuvent utiliser des métadonnées contextuelles :

- Recommandé pour vous(domaine ECOMMERCE)
- Les meilleurs choix pour vous(domaine VIDEO_ON_DEMAND)

Pour les ressources personnalisées, les recettes qui utilisent des métadonnées contextuelles sont les suivantes :

- Personnalisation de l'utilisateur-v2 et Personnalisation par l'utilisateur
- Ranking-V2 personnalisé et Personalized-Ranking

Pour plus d'informations sur l'inclusion du contexte lorsque vous recevez des recommandations, consultezAméliorer la pertinence des recommandations grâce aux métadonnées contextuelles. Pour un exemple complet illustrant comment utiliser les métadonnées contextuelles, consultez le billet de blog sur le AWS Machine Learning suivant : Augmenter la pertinence de vos recommandations Amazon Personalize en tirant parti des informations contextuelles.

Données sur les impressions

Les impressions sont des listes d'éléments visibles par un utilisateur lorsqu'il interagissait avec un élément en particulier (par exemple, en cliquant ou en regardant). Si vous utilisez un cas d'utilisation de domaine qui fournit la personnalisation ou la Personnalisation par l'utilisateur recette, Amazon Personalize peut utiliser les données d'impressions pour guider l'exploration.

Dans le cadre de l'exploration, les recommandations incluent certains éléments ou actions qui sont généralement moins susceptibles d'être recommandés à l'utilisateur, tels que de nouveaux éléments ou actions, des éléments ou des actions avec peu d'interactions, ou des éléments ou actions moins pertinents pour l'utilisateur en raison de son comportement antérieur. Plus un article apparaît fréquemment dans les données d'impressions, moins il est probable qu'Amazon Personalize l'inclut dans l'exploration.

Lorsque vous créez un recommandeur ou une solution, Amazon Personalize exclut toujours les données relatives aux impressions de la formation. Cela est dû au fait qu'Amazon Personalize n'entraîne pas vos modèles à l'aide des données d'impressions. Il l'utilise plutôt lorsque vous recevez des recommandations pour guider l'exploration de l'utilisateur.

Les valeurs d'impression peuvent comporter au maximum 1 000 caractères (y compris la barre verticale). Pour les groupes de jeux de données de domaine, les cas d'utilisation des recommandations suivants peuvent utiliser des données d'impressions :

- Recommandé pour vous(domaine ECOMMERCE)
- Les meilleurs choix pour vous(domaine VIDEO_ON_DEMAND)

Pour plus d'informations sur l'exploration, voirExploration. Amazon Personalize peut modéliser deux types d'impressions : Impressions implicites etImpressions explicites.

Impressions explicites

Les impressions explicites sont des impressions que vous enregistrez et envoyez manuellement à Amazon Personalize. Utilisez des impressions explicites pour manipuler les résultats d'Amazon Personalize. L'ordre des articles n'a aucun impact.

Par exemple, vous pouvez avoir une application d'achat qui fournit des recommandations pour les chaussures. Si vous ne recommandez que des chaussures actuellement en stock, vous pouvez spécifier ces articles en utilisant des impressions explicites. Votre flux de travail de recommandation utilisant des impressions explicites peut être le suivant :

1. Vous demandez des recommandations pour l'un de vos utilisateurs à l'aide de l'the section called “GetRecommendations”API Amazon Personalize.
2. Amazon Personalize génère des recommandations pour l'utilisateur utilisant votre modèle (version de la solution) et les renvoie dans la réponse de l'API.
3. Vous ne montrez à l'utilisateur que les chaussures recommandées en stock.
4. Pour l'importation de données incrémentielle en temps réel, lorsque votre utilisateur interagit avec (par exemple, clique) avec une paire de chaussures, vous enregistrez son choix lors d'un appel à l'PutEventsAPI et vous listez les articles recommandés qui sont en stock dans le `impression` paramètre. Pour un exemple de code, voirEnregistrement des événements d'interaction avec les objets à l'aide des données d'impressions.

   Pour importer des impressions dans les données historiques des interactions entre articles, vous pouvez répertorier les impressions explicites dans votre fichier csv et séparer chaque élément par un caractère « | ». Les caractères de la barre verticale sont pris en compte dans la limite de 1 000 caractères. Pour obtenir un exemple, consultez Formatage d'impressions explicites.

5. Amazon Personalize utilise les données d'impression pour guider l'exploration, et les recommandations futures incluent de nouvelles chaussures avec moins de données d'interactions ou de pertinence.

Formatage d'impressions explicites

Pour inclure des impressions explicites dans votre fichier CSV, ajoutez une colonne IMPRESSION. Pour chaque interaction avec un objet, ajoutez une liste d'ItemID séparés par une barre verticale, le caractère « | ». Les caractères de la barre verticale sont pris en compte dans la limite de 1 000 caractères pour les données d'impressions. Si vous incluez des impressions explicites dans le PutEvents fonctionnement, vous spécifiez les éléments dans un tableau de chaînes.

Ce qui suit est un court extrait d'un fichier CSV qui inclut des impressions explicites dans la `IMPRESSION` colonne.

EVENT_TYPE	IMPRESSION	ITEM_ID	TIMESTAMP	USER_ID
cliquez sur	73|70|17|95|96	73	1586731606	UTILISATEUR_1
cliquez sur	35|82|78|57|20|63|1|90|76|75|49|71|26|24|25|6	35	1586735164	UTILISATEUR_2
...	...	...	...	...

L'application a affiché `USER_1` les éléments utilisateur `73`,`70`,`17`,`95`, 96 et l'utilisateur a finalement choisi l'élément`73`. Lorsque vous créez une nouvelle version de solution basée sur ces données`70`, les éléments `17`,`95`, et 96 seront moins fréquemment recommandés aux utilisateurs`USER_1`.

Impressions implicites

Les impressions implicites sont les recommandations, extraites d'Amazon Personalize, que vous montrez à l'utilisateur. Votre fichier CSV n'a pas besoin d'inclure les colonnes IMPRESSION ou RECOMMENDATION_ID pour utiliser des impressions implicites. Au lieu de cela, vous incluez le `RecommendationId` (renvoyé par les GetPersonalizedRanking opérations GetRecommendations et) dans les PutEvents demandes. Amazon Personalize calcule les impressions implicites en fonction de vos données de recommandation.

Par exemple, vous pouvez avoir une application qui fournit des recommandations pour le streaming vidéo. Votre flux de travail de recommandation utilisant des impressions implicites peut être le suivant :

1. Vous demandez des recommandations vidéo à l'un de vos utilisateurs à l'aide de l'opération Amazon Personalize the section called “GetRecommendations” API.
2. Amazon Personalize génère des recommandations pour l'utilisateur utilisant votre modèle (version de la solution) et les renvoie avec un `recommendationId` dans la réponse de l'API.
3. Vous montrez les recommandations vidéo à votre utilisateur dans votre application.
4. Lorsque votre utilisateur interagit avec une vidéo (par exemple, clique), enregistrez le choix dans un appel à l'PutEventsAPI et incluez-le en `recommendationId` tant que paramètre. Pour un exemple de code, voirEnregistrement des événements d'interaction avec les objets à l'aide des données d'impressions.
5. Amazon Personalize utilise les données d'impression `recommendationId` à partir des recommandations vidéo précédentes, puis utilise les données d'impression pour guider l'exploration, les recommandations futures incluant de nouvelles vidéos avec moins de données d'interactions ou de pertinence.

   Pour plus d'informations sur l'enregistrement d'événements avec des données d'impression implicites, consultezEnregistrement des événements d'interaction avec les objets à l'aide des données d'impressions.

Exemple de données d'interactions

Les données d'interactions suivantes représentent l'historique de l'activité des utilisateurs sur un site Web de streaming vidéo. Vous pouvez utiliser les données pour entraîner un modèle qui fournit des recommandations de films en fonction des données d'interaction des utilisateurs. Notez que certaines valeurs de EVENT_VALUE sont nulles.

```
USER_ID,ITEM_ID,EVENT_TYPE,EVENT_VALUE,TIMESTAMP
196,242,watch,.50,881250949
186,302,watch,.75,891717742
22,377,click,,878887116
244,51,click,,880606923
166,346,watch,.50,886397596
298,474,watch,.25,884182806
115,265,click,,881171488
253,465,watch,.50,891628467
```

```
305,451,watch,.75,886324817
```

Amazon Personalize nécessite la `TIMESTAMP` colonne `USER_ID``ITEM_ID`, et. `USER_ID`Dest l'identifiant d'un utilisateur de votre application. `ITEM_ID`Dest l'identifiant d'un film. `EVENT_TYPE`et `EVENT_VALUE` sont les identifiants des interactions des utilisateurs. Dans les exemples de données, les événements sont `click` des événements `watch` et les valeurs sont le pourcentage d'une vidéo regardée par un utilisateur. Cela `TIMESTAMP` représente l'époque Unix à laquelle l'achat du film a eu lieu.

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize. Voici à quoi ressemblerait le fichier de schéma JSON pour les exemples de données.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    { "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": "float"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
```

```
}
```

Préparation des métadonnées des éléments pour la formation

Les métadonnées des articles incluent des données numériques et catégoriques sur les éléments avec lesquels vos utilisateurs interagissent. Les métadonnées des articles incluent l'horodatage de création, le prix, le genre, la description et la disponibilité. Vous importez les métadonnées relatives à vos articles dans un ensemble de données Amazon Personalize Items.

Selon le cas d'utilisation de votre domaine ou votre recette personnalisée, les métadonnées des articles peuvent aider Amazon Personalize à recommander des articles plus pertinents aux utilisateurs, à prévoir des articles similaires avec plus de précision ou à recommander des segments d'utilisateurs plus pertinents. Et cela peut aider Amazon Personalize à proposer de nouveaux articles dans ses recommandations. Les métadonnées des éléments sont obligatoires pour certains cas d'utilisation du domaine et facultatives pour toutes les recettes personnalisées. Pour plus d'informations, consultez les exigences en matière de données pour le cas d'utilisation ou la recette de votre domaine dansAdapter votre cas d'utilisation aux ressources Amazon Personalize.

Lors de la formation, Amazon Personalize n'utilise pas de chaînes de données non catégoriques, telles que les titres des articles ou les données relatives à l'auteur. Cependant, l'importation de ces données peut toujours améliorer les recommandations. Pour de plus amples informations, veuillez consulter Données de chaîne non catégoriques.

Le nombre maximum d'éléments pris en compte par Amazon Personalize pendant la formation dépend de votre cas d'utilisation ou de votre recette. Seuls les éléments pris en compte lors de la formation peuvent apparaître dans les recommandations.

- Pour User-Personalization-v 2 ou Personalized-Ranking-v 2, le nombre maximum d'objets pris en compte par un modèle pendant l'entraînement est de 5 millions. Ces éléments proviennent à la fois du jeu de données Items et du jeu de données des interactions Item.
- Pour tous les cas d'utilisation du domaine et les recettes personnalisées autres que User-Personalization-v 2 et Personalized-Ranking-v 2, le nombre maximum d'éléments pris en compte par un modèle lors de la formation et de la génération de recommandations est de 750 000.

Pour tous les cas d'utilisation du domaine et les recettes personnalisées, les données relatives aux articles en vrac doivent figurer dans un fichier CSV. Chaque ligne du fichier doit représenter un élément unique. Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour

de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Les sections suivantes fournissent plus d'informations sur la façon de préparer les métadonnées de vos articles pour Amazon Personalize. Pour les directives relatives au format des données en masse pour tous les types de données, consultez les directives relatives au format des données en masse

Rubriques

- Exigences relatives aux données relatives aux articles
- Données d'horodatage de création
- Métadonnées catégoriques
- Métadonnées de texte non structurées
- Données numériques
- Données de chaîne non catégoriques
- Exemple de métadonnées d'éléments

Exigences relatives aux données relatives aux articles

Voici les exigences relatives aux métadonnées des articles pour Amazon Personalize.

Si vous n'êtes pas sûr de disposer de suffisamment de données ou si vous avez des questions quant à leur qualité, vous pouvez importer vos données dans un ensemble de données Amazon Personalize et utiliser Amazon Personalize pour les analyser. Pour de plus amples informations, veuillez consulter Analyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

- Pour tous les cas d'utilisation du domaine et les recettes personnalisées, vous devez disposer d'une colonne ITEM_ID qui stocke l'identifiant unique de chaque élément. Chaque article doit avoir un numéro d'article. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- Pour les recettes personnalisées, vos données doivent comporter au moins une chaîne catégorique ou une colonne de métadonnées numériques. Les colonnes de métadonnées des éléments peuvent inclure empty/null des valeurs. Nous recommandons que ces colonnes soient remplies à au moins 70 %.
- Pour les cas d'utilisation de domaines, les colonnes requises dépendent de votre domaine. Pour plus d’informations, consultez Exigences relatives au domaine VIDEO_ON_DEMAND ou Exigences relatives au domaine ECOMMERCE.

- Le nombre maximum de colonnes de métadonnées est de 100.

Exigences relatives au domaine VIDEO_ON_DEMAND

Les métadonnées d'un élément sont requises dans certains cas d'utilisation (voirCas d'utilisation de VIDEO_ON_DEMAND). Lorsque cela est facultatif, nous vous recommandons tout de même d'importer les métadonnées des éléments pour obtenir les recommandations les plus pertinentes. Si vous importez des métadonnées d'articles, vos données doivent inclure les colonnes suivantes :

- ITEM_ID
- GENRES (catégoriques`string`)
- CREATION_TIMESTAMP (au format Unix Epoch Time)

Vous trouverez ci-dessous la liste des colonnes supplémentaires recommandées et leurs types obligatoires. Le `null` type indique que des valeurs peuvent être manquantes dans la colonne. Nous recommandons que ces colonnes soient remplies à au moins 70 %. L'inclusion de ces colonnes peut améliorer les recommandations.

- PRIX (flottant)
- DURÉE (flottant)
- GENRE_L2 (catégorique,) `string` `null`
- GENRE_L3 (catégorique,) `string` `null`
- NOTE MOYENNE (,) `float` `null`
- DESCRIPTION DU PRODUIT (`string`textuelle,) `null`
- CONTENT_OWNER (categorical`string`,`null`) — La société propriétaire de la vidéo. Par exemple, les valeurs peuvent être HBO, Paramount et NBC.
- CONTENT_CLASSIFICATION (catégorique`string`,`null`) — Évaluation du contenu. Par exemple, les valeurs peuvent être G, PG, PG-13, R, NC-17 et rated.

Exigences relatives au domaine ECOMMERCE

Les métadonnées des articles sont facultatives pour tous les cas d'utilisation du commerce électronique. Si vous avez des données sur les articles, nous vous recommandons de les importer pour obtenir les recommandations les plus pertinentes. Si vous importez des métadonnées d'articles, vos données doivent comporter les colonnes suivantes :

- ITEM_ID
- PRIX (`float`)
- CATEGORY_L1 (catégorique`string`) — Pour plus d'informations sur le formatage des données catégorielles, consultez. Métadonnées catégoriques

Vous trouverez ci-dessous la liste des colonnes supplémentaires recommandées et leurs types obligatoires. Le `null` type indique que des valeurs peuvent être manquantes dans la colonne. Nous recommandons que ces colonnes soient remplies à au moins 70 %. L'inclusion de ces colonnes peut améliorer les recommandations.

- CATEGORY_L2 (catégorique,) `string` `null`
- CATEGORY_L3 (catégorique,) `string` `null`
- DESCRIPTION DU PRODUIT (`string`textuelle,) `null`
- HORODATAGE DE CRÉATION () `float`
- AGE_GROUP (catégorique`string`,`null`) — Le groupe d'âge auquel l'article est destiné. Les valeurs peuvent être les nouveau-nés, les nourrissons, les enfants et les adultes.
- ADULTE (catégorique`string`,`null`) — Indique si l'article est réservé aux adultes, comme l'alcool. Les valeurs peuvent être oui ou non.
- GENRE (catégorique`string`,`null`) — Le sexe auquel l'article est destiné. Les valeurs peuvent être masculines, féminines ou unisexes.

Données d'horodatage de création

Les données d'horodatage de création doivent être au format Unix Epoch Time en secondes. Par exemple, l'horodatage Epoch en secondes pour la date du 31 juillet 2020 est 1596238243. Pour convertir les dates en horodatages d'époque Unix, utilisez un convertisseur Epoch, un convertisseur d'horodatage Unix.

Amazon Personalize utilise les données d'horodatage de création (au format Unix Epoch, en secondes) pour calculer l'âge d'un article et ajuster les recommandations en conséquence.

Si les données d'horodatage de création sont manquantes pour un ou plusieurs articles, Amazon Personalize déduit ces informations à partir des données d'interaction, le cas échéant, et utilise l'horodatage des données d'interaction les plus anciennes de l'article comme date de création de l'article. Si un article ne contient aucune donnée d'interaction, son horodatage de création est défini

comme l'horodatage de la dernière interaction du set d'entraînement et Amazon Personalize le considère comme un nouvel article.

Métadonnées catégoriques

Avec certaines recettes et tous les cas d'utilisation de domaines, Amazon Personalize utilise des métadonnées catégoriques, telles que le genre ou la couleur d'un article, pour identifier les modèles sous-jacents qui révèlent les articles les plus pertinents pour vos utilisateurs. Vous définissez votre propre plage de valeurs en fonction de votre cas d'utilisation. Les métadonnées catégorielles peuvent être rédigées dans n'importe quelle langue.

Pour les articles comportant plusieurs catégories, séparez chaque valeur par la barre verticale « | ». Par exemple, pour un champ GENRES, les données relatives à un élément peuvent être`Action|Crime|Biopic`. Si vous disposez de plusieurs niveaux de données catégorielles et que certains éléments comportent plusieurs catégories pour chaque niveau de la hiérarchie, utilisez une colonne distincte pour chaque niveau et ajoutez un indicateur de niveau après le nom de chaque champ : GENRES, GENRE_L2, GENRE_L3. Cela vous permet de filtrer les recommandations en fonction de sous-catégories, même si un article appartient à plusieurs catégories à plusieurs niveaux (pour plus d'informations sur la création et l'utilisation de filtres, voirRecommandations de filtrage et segments d'utilisateurs). Par exemple, une vidéo peut contenir les données suivantes pour chaque niveau de catégorie :

- GENRES : Action|Aventure
- GENRE_L2 : Crime | Western
- GENRE_L3 : Biopic

Dans cet exemple, la vidéo se trouve dans la hiérarchie action > crime > biopic et dans la hiérarchie aventure > western > biopic. Nous vous recommandons de n'utiliser que le niveau L3, mais vous pouvez utiliser d'autres niveaux si nécessaire.

Les valeurs catégorielles peuvent comporter un maximum de 1 000 caractères. Si vous avez un élément avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera. Nous recommandons que les colonnes catégorielles comportent au maximum 1 000 valeurs possibles. L'importation de données catégorielles comportant davantage de valeurs peut avoir un impact négatif sur les recommandations. Les éléments suivants peuvent vous aider à réduire le nombre de valeurs possibles pour une colonne catégorielle :

- Assurez-vous que les valeurs respectent une convention de dénomination cohérente et vérifiez qu'il n'y a pas de fautes de frappe. Par exemple, utilisez « Chaussures pour hommes » plutôt que d'avoir un mélange de « chaussures pour hommes », « chaussures pour hommes » et « chaussures pour hommes ».
- Consolidez les catégories similaires qui utilisent des termes légèrement différents faisant référence à la même catégorie sous-jacente, comme « Chaussures » et « Espadrilles ».
- Si vos données ont une structure hiérarchique, dans laquelle des catégories plus larges (comme « Chaussures ») contiennent des sous-catégories plus spécifiques (telles que « Chaussures pour hommes », « Chaussures pour femmes », « Chaussures pour enfants »), utilisez une colonne distincte pour chaque niveau et ajoutez un indicateur de niveau après le nom de chaque champ. Par exemple, CATEGORY_1, CATEGORY_2 et CATEGORY_3. Cela permet de réduire les catégories ambiguës ou qui se chevauchent.

Avec toutes les recettes et tous les domaines, vous pouvez importer des données catégorielles et les utiliser pour filtrer les recommandations en fonction des attributs d'un article. Pour plus d'informations sur les recommandations de filtrage, consultezRecommandations de filtrage et segments d'utilisateurs.

Métadonnées de texte non structurées

Avec certaines recettes et certains domaines, Amazon Personalize peut extraire des informations pertinentes à partir de métadonnées de texte non structurées, telles que des descriptions de produits, des critiques de produits ou des synopsis de films. Amazon Personalize utilise du texte non structuré pour identifier les articles pertinents pour vos utilisateurs, en particulier lorsque les articles sont nouveaux ou contiennent moins de données d'interactions. Vous pouvez ajouter au plus un champ textuel. Incluez des données textuelles non structurées dans votre jeu de données d'articles afin d'augmenter les taux de clics et les taux de conversation pour les nouveaux articles de votre catalogue.

Lorsque vous préparez vos métadonnées de texte non structuré, placez le texte entre guillemets et supprimez tout nouveau caractère de ligne. Utilisez le \ caractère pour éviter les guillemets ou les caractères \ dans vos données. Amazon Personalize tronque les champs de texte à la limite de caractères. Assurez-vous que les informations les plus pertinentes du texte se trouvent au début du champ.

Les valeurs de texte non structurées peuvent comporter au maximum 20 000 caractères dans toutes les langues sauf le chinois et le japonais. Pour le chinois et le japonais, vous pouvez compter au

maximum 7 000 caractères. Amazon Personalize tronque les valeurs qui dépassent la limite de caractères jusqu'à la limite de caractères.

Vous pouvez soumettre des éléments de texte non structurés dans plusieurs langues, mais le texte de chaque élément ne doit être rédigé que dans une seule langue. Le texte peut être rédigé dans les langues suivantes :

- Chinois (simplifié)
- Chinois (Traditionnel)
- Anglais
- Français
- Allemand
- Japonais
- Portugais
- Espagnol

Données numériques

Amazon Personalize peut utiliser les métadonnées numériques des articles, telles que le prix ou la durée de la vidéo, afin de générer des recommandations plus pertinentes pour les utilisateurs. Ces données numériques peuvent être représentées sous forme de nombres entiers ou de valeurs décimales.

Si vous utilisez des Personnalisation par l'utilisateur recettes Personalized-Ranking personnalisées, vous pouvez optimiser une solution Amazon Personalize pour un objectif lié aux métadonnées d'un article, en plus d'une pertinence maximale, telle que la maximisation des revenus. Lorsque vous configurez votre solution, vous choisissez la colonne de métadonnées numériques de votre jeu de données Items qui est liée à votre objectif. Par exemple, vous pouvez choisir une colonne VIDEO_LENGTH pour maximiser les minutes de diffusion ou une colonne PRICE pour maximiser les revenus.

Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.

Données de chaîne non catégoriques

À l'exception des articles IDs, Amazon Personalize n'utilise pas de chaînes non textuelles non catégoriques lors de la formation, telles que les titres des articles ou les données relatives à l'auteur. Amazon Personalize peut toutefois l'utiliser avec les fonctionnalités suivantes. Les valeurs non catégoriques peuvent comporter un maximum de 1 000 caractères.

- Amazon Personalize peut inclure les métadonnées des articles dans les recommandations, y compris des valeurs de chaîne non catégoriques. Vous pouvez utiliser des métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant le nom du réalisateur à un carrousel de recommandations de films. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.
- Si vous en utilisezObjets similaires, vous pouvez générer des recommandations par lots avec des thèmes. Lorsque vous générez des recommandations par lots avec des thèmes, vous devez spécifier une colonne de nom d'article dans le travail d'inférence par lots. Pour de plus amples informations, veuillez consulter Recommandations par lots avec des thèmes issus du générateur de contenu.
- Vous pouvez créer des filtres pour inclure ou supprimer des éléments des recommandations sur la base de données de chaîne non catégoriques. Pour plus d'informations sur les filtres, consultez Recommandations de filtrage et segments d'utilisateurs.

Exemple de métadonnées d'éléments

Les premières lignes des métadonnées d'un film dans un fichier CSV peuvent ressembler à ce qui suit.

```
ITEM_ID,GENRES,CREATION_TIMESTAMP,DESCRIPTION
1,Adventure|Animation|Children|Comedy|Fantasy,1570003267,"This is an animated movie
 that features action, comedy, and fantasy. Audience is children. This movie was
 released in 2004."
2,Adventure|Children|Fantasy,1571730101,"This is an adventure movie with elements of
 fantasy. Audience is children. This movie was release in 2010."
3,Comedy|Romance,1560515629,"This is a romantic comedy. The movie was released in 1999.
 Audience is young women."
4,Comedy|Drama|Romance,1581670067,"This movie includes elements of both comedy and
 drama as well as romance. This movie was released in 2020."
...
...
```

La `ITEM_ID` colonne est obligatoire et contient des identifiants uniques pour chaque article individuel. La `GENRE` colonne stocke les métadonnées catégorielles pour chaque film et la `DESCRIPTION` colonne contient des métadonnées textuelles non structurées. La `CREATION_TIMESTAMP` colonne stocke l'heure de création de chaque élément au format Unix Epoch Time en secondes.

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize. Voici à quoi ressemblerait le fichier de schéma JSON pour les exemples de données ci-dessus.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "DESCRIPTION",
      "type": [
        "null",
        "string"
      ],
      "textual": true
    }
  ],
  "version": "1.0"
```

```
}
```

Préparation des métadonnées utilisateur pour la formation

Les données utilisateur que vous pouvez importer dans Amazon Personalize incluent des données numériques, telles que l'âge de l'utilisateur, et des métadonnées catégoriques, telles que le sexe ou le niveau de fidélité. Vous importez les métadonnées relatives à vos utilisateurs dans un ensemble de données Amazon Personalize Users.

Selon le cas d'utilisation de votre domaine ou votre recette personnalisée, les métadonnées utilisateur peuvent aider Amazon Personalize à recommander des articles plus pertinents aux utilisateurs ou à recommander des segments d'utilisateurs plus pertinents. Et après l'entraînement, cela peut aider votre modèle à recommander des éléments aux utilisateurs sans aucune donnée d'interaction. Pour plus d'informations sur les cas d'utilisation ou les recettes qui utilisent les métadonnées utilisateur, consultez les exigences en matière de données pour le cas d'utilisation ou la recette de votre domaine dansAdapter votre cas d'utilisation aux ressources Amazon Personalize.

Lors de la formation, Amazon Personalize n'utilise pas de chaînes de données utilisateur non catégoriques, telles que les noms des utilisateurs, les mots clés relatifs à l'utilisateur ou les tags. Toutefois, l'importation de ces données peut tout de même améliorer les recommandations. Pour de plus amples informations, veuillez consulter Données de chaîne non catégoriques.

Pour tous les cas d'utilisation du domaine et les recettes personnalisées, vos données utilisateur groupées doivent figurer dans un fichier CSV. Chaque ligne du fichier doit représenter un utilisateur unique. Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Les sections suivantes fournissent plus d'informations sur la façon de préparer vos données utilisateur pour Amazon Personalize. Pour les directives relatives au format des données en masse pour tous les types de données, consultez les directives relatives au format des données en masse

Rubriques

- Exigences relatives aux données utilisateur
- Métadonnées catégoriques
- Données de chaîne non catégoriques
- Exemple de métadonnées utilisateur

Exigences relatives aux données utilisateur

Voici les exigences relatives aux données utilisateur pour Amazon Personalize. Vous êtes libre d'ajouter des colonnes personnalisées supplémentaires en fonction de votre cas d'utilisation et de vos données.

- Vos données doivent comporter une colonne USER_ID qui stocke l'identifiant unique de chaque utilisateur. Chaque utilisateur doit avoir un nom d'utilisateur. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- Vos données doivent comporter au moins une chaîne catégorique ou une colonne de métadonnées numériques. Les colonnes de métadonnées utilisateur peuvent inclure empty/null des valeurs pour certains utilisateurs. Nous recommandons que ces colonnes soient remplies à au moins 70 %.
- Le nombre maximum de colonnes de métadonnées est de 25.

Si vous n'êtes pas sûr de disposer de suffisamment de données ou si vous avez des questions quant à leur qualité, vous pouvez importer vos données dans un ensemble de données Amazon Personalize et utiliser Amazon Personalize pour les analyser. Pour de plus amples informations, veuillez consulter Analyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

Métadonnées catégoriques

Avec certaines recettes et tous les cas d'utilisation de domaines, Amazon Personalize utilise des métadonnées catégoriques, telles que le sexe, les intérêts ou le statut de membre d'un utilisateur, pour identifier les modèles sous-jacents qui révèlent les éléments les plus pertinents pour vos utilisateurs. Vous définissez votre propre plage de valeurs en fonction de votre cas d'utilisation. Les métadonnées catégorielles peuvent être rédigées dans n'importe quelle langue.

Pour les utilisateurs appartenant à plusieurs catégories, séparez chaque valeur par la barre verticale « | ». Par exemple, pour un champ INTÉRÊTS, vos données relatives à un utilisateur peuvent être`Movies|TV Shows|Music`.

Avec toutes les recettes et tous les domaines, vous pouvez importer des métadonnées catégorielles et les utiliser pour filtrer les recommandations en fonction des attributs de l'utilisateur. Pour plus d'informations sur les recommandations de filtrage, voirRecommandations de filtrage et segments d'utilisateurs.

Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si un utilisateur possède une valeur catégorielle de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Données de chaîne non catégoriques

À l'exception de l'utilisateur IDs, Amazon Personalize n'utilise pas de chaînes de données non catégoriques lors de la formation, telles que les noms d'utilisateur, les mots clés relatifs à l'utilisateur ou les tags. Amazon Personalize peut toutefois l'utiliser pour filtrer les recommandations. Vous pouvez créer des filtres pour inclure ou supprimer des éléments dans les recommandations en fonction de chaînes de données non catégoriques concernant l'utilisateur pour lequel vous recevez des recommandations (le CurrentUser). Pour plus d'informations sur les filtres, consultez Recommandations de filtrage et segments d'utilisateurs. Les valeurs non catégoriques peuvent comporter un maximum de 1 000 caractères.

Exemple de métadonnées utilisateur

Les premières lignes de métadonnées utilisateur d'un fichier CSV peuvent ressembler à ce qui suit.

```
USER_ID,AGE,GENDER,INTEREST
5,34,Male,hiking
6,56,Female,music
8,65,Male,movies|TV shows|music
...
...
```

La `USER_ID` colonne est obligatoire et stocke des identifiants uniques pour chaque utilisateur individuel. La `AGE` colonne contient des métadonnées numériques. Les `INTEREST` colonnes `GENDER` et stockent les métadonnées catégoriques pour chaque utilisateur.

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize. Voici à quoi ressemblerait le fichier de schéma JSON pour les exemples de données ci-dessus.

```
{
  "type": "record",
  "name": "Users",
```

```
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "AGE",
          "type": "int"
      },
      {
          "name": "GENDER",
          "type": "string",
          "categorical": true
      },
      {
          "name": "INTEREST",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

Préparation des métadonnées d'action pour la formation

Une action est une activité génératrice d'engagement ou de revenus que vous souhaiterez peut-être recommander à vos utilisateurs. Les actions peuvent inclure l'installation de votre application mobile, la création d'un profil de membre, l'adhésion à votre programme de fidélité ou l'inscription à des e-mails promotionnels. Vous importez les données relatives à vos actions dans un ensemble de données Amazon Personalize Actions. Les exemples de données relatives à une action incluent l'identifiant unique de l'action, la valeur estimée de l'action ou l'horodatage d'expiration de l'action.

Si vous utilisez Next-Best-Action, vous devez importer les métadonnées des actions. Avec cette recette, Amazon Personalize prédit la meilleure action suivante à partir des actions que vous importez dans votre ensemble de données Actions. Aucune autre recette ou cas d'utilisation n'utilise de métadonnées d'action. Vous ne pouvez pas créer de jeu de données Actions dans un groupe de jeux de données de domaine.

Lors de la formation, Amazon Personalize n'utilise pas de données d'actions de chaîne non catégoriques, telles que les titres ou les tags des actions. Cependant, l'importation de ces données

peut toujours améliorer les recommandations. Pour de plus amples informations, veuillez consulter Données de chaîne non catégoriques.

Les données relatives à vos actions groupées doivent se trouver dans un fichier CSV. Chaque ligne du fichier doit représenter une action unique. Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Les sections suivantes fournissent plus d'informations sur la façon de préparer les métadonnées de vos actions pour Amazon Personalize. Pour les directives relatives au format des données en masse pour tous les types de données, consultez les directives relatives au format des données en masse

Rubriques

- Exigences relatives aux données relatives aux actions
- Données d'horodatage d'expiration des actions
- Répéter les données de fréquence
- Données de valeur
- Données d'horodatage de création
- Métadonnées catégoriques
- Données de chaîne non catégoriques
- Exemple de métadonnées d'actions

Exigences relatives aux données relatives aux actions

Les données d'action requises pour Amazon Personalize sont les suivantes.

- Vous devez disposer d'une colonne ACTION_ID qui stocke l'identifiant unique de chaque action. Chaque action doit avoir un identifiant d'article. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- Vos données doivent comporter au moins une chaîne catégorique ou une colonne de métadonnées numériques. Les colonnes de métadonnées d'action peuvent inclure empty/null des valeurs. Nous recommandons que ces colonnes soient remplies à au moins 70 %.
- Au cours de la formation des modèles, Amazon Personalize prend en compte un maximum de 1 000 actions. Si vous importez plus de 1 000 actions, Amazon Personalize décide quelles actions

inclure dans la formation, la priorité étant donnée aux nouvelles actions (actions que vous avez récemment ajoutées sans interaction) et aux actions existantes avec des données d'interactions récentes.

- Le nombre maximum de colonnes est de 10.

Données d'horodatage d'expiration des actions

L'horodatage d'expiration d'une action indique la date à laquelle une action n'est plus valide. Vous fournissez les données d'horodatage d'expiration des actions au format Unix Epoch Time, en secondes. Si une action a expiré, Amazon Personalize ne l'inclura pas dans les recommandations.

Spécifiez un horodatage d'expiration pour vos actions si vous souhaitez limiter leur apparition dans les recommandations à une certaine période. Par exemple, vous pouvez avoir une application qui lance une campagne d'adhésion pendant un mois donné. Vous pouvez définir un horodatage d'expiration pour l'action d'inscription à la fin du mois en question. Amazon Personalize cesse automatiquement de recommander cette action lorsque cette date est atteinte.

Si vous définissez l'horodatage d'expiration à une date antérieure pour une nouvelle action, ou si vous mettez à jour l'horodatage d'une action à une date antérieure, la suppression de l'action des recommandations peut prendre jusqu'à 2 heures.

Répéter les données de fréquence

Les données de fréquence de répétition indiquent le nombre de jours qu'Amazon Personalize doit attendre pour recommander une action particulière après qu'un utilisateur interagisse avec celle-ci, en fonction de l'historique de l'utilisateur dans votre ensemble de données d'interactions avec les actions. Vous spécifiez la fréquence de répétition d'une action en jours, avec un maximum de 30 jours.

Par exemple, vous pouvez avoir une application de commerce électronique dans laquelle chaque utilisateur crée un compte et un profil. Si vous avez une `complete profile` action et que vous souhaitez attendre une semaine après qu'un utilisateur interagisse avec elle avant de la recommander à nouveau, vous devez spécifier une durée de 7 jours pour l'action. `REPEAT_FREQENCY` Au bout de 7 jours, Amazon Personalize commence à envisager les mesures à prendre pour obtenir des recommandations.

Si vous ne fournissez pas de fréquence de répétition pour une action, Amazon Personalize ne fixera aucune limite quant au nombre de fois qu'elle apparaît dans les recommandations.

Données de valeur

Les données de valeur sont la valeur commerciale ou l'importance de chaque action. Une action `value` peut être comprise entre 1 et 10, 10 étant l'action la plus précieuse de votre ensemble de données.

Par exemple, vous pouvez avoir deux actions, l'une pour vous inscrire à votre abonnement de base et l'autre pour vous inscrire à votre service premium. Pour le service de base, vous pouvez spécifier une valeur de `5` et pour la prime, une valeur de`10`.

Amazon Personalize utilise les données de valeur comme entrée pour déterminer la meilleure action à recommander à vos utilisateurs. Par exemple, si un utilisateur est également susceptible d'effectuer une action ou une autre, Amazon Personalize classe l'action ayant la valeur la plus élevée dans les recommandations.

Données d'horodatage de création

Amazon Personalize utilise les données d'horodatage de création (au format Unix Epoch, en secondes) pour calculer l'âge d'une action et ajuster les recommandations en conséquence.

Si vous ne disposez pas de données d'horodatage de création, Amazon Personalize déduit ces informations à partir des données d'interaction entre les actions. Il utilise l'horodatage des données d'interaction les plus anciennes de l'action comme horodatage de création de l'action. Si une action ne contient aucune donnée d'interaction, son horodatage de création est défini comme l'horodatage de la dernière interaction du set d'entraînement, et Amazon Personalize considère qu'il s'agit d'une nouvelle action.

Métadonnées catégoriques

Amazon Personalize utilise des métadonnées catégoriques concernant les actions, telles que le caractère saisonnier ou l'exclusivité des actions, pour identifier les modèles sous-jacents qui révèlent les meilleures actions pour vos utilisateurs. Vous définissez votre propre plage de valeurs en fonction de votre cas d'utilisation. Les métadonnées catégorielles peuvent être rédigées dans n'importe quelle langue.

Vous pouvez importer des données catégorielles et les utiliser pour filtrer les recommandations en fonction des attributs d'une action. Pour plus d'informations sur les recommandations de filtrage, consultezRecommandations de filtrage et segments d'utilisateurs.

Les valeurs catégorielles peuvent comporter un maximum de 1 000 caractères. Si vous avez une action avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Données de chaîne non catégoriques

À l'exception de l'action IDs, Amazon Personalize n'utilise pas de données de chaîne non catégoriques lors de l'entraînement, telles que le nom d'une action, les mots clés relatifs à l'action ou les balises. Amazon Personalize peut toutefois l'utiliser pour filtrer les recommandations. Vous pouvez créer des filtres pour inclure ou supprimer des actions dans les recommandations en fonction de données de chaîne non catégoriques. Pour plus d'informations sur les filtres, consultez Recommandations de filtrage et segments d'utilisateurs. Les valeurs non catégoriques peuvent comporter un maximum de 1 000 caractères.

Exemple de métadonnées d'actions

Les métadonnées des premières lignes d'action d'un fichier CSV peuvent ressembler à ce qui suit.

```
ACTION_ID,VALUE,MEMBERSHIP_LEVEL,CREATION_TIMESTAMP,REPEAT_FREQUENCY
1,10,Deluxe|Premium,1510003267,7
2,5,Basic,1580003267,7
3,5,Preview,1590003267,3
4,10,Deluxe|Platinum,1560003267,4
...
...
```

La colonne `ACTION_ID` est obligatoire. La `MEMBERSHIP_LEVEL` colonne est un champ de chaîne catégorique. Les `REPEAT_FREQUENCY` champs `VALUECREATION_TIMESTAMP`, et sont des mots clés réservés avec les types requis.

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize. Voici à quoi ressemblerait le fichier de schéma JSON pour les exemples de données ci-dessus.

```
{
  "type": "record",
  "name": "Actions",
  "namespace": "com.amazonaws.personalize.schema",
```

```
  "fields": [
    {
      "name": "ACTION_ID",
      "type": "string"
    },
    {
      "name": "VALUE",
      "type": [
        "null",
        "long"
      ]
    },

    {
      "name": "MEMBERSHIP_LEVEL",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
    },

    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "REPEAT_FREQUENCY",
      "type": [
        "long",
        "null"
      ]
    }
  ],
  "version": "1.0"
}
```

Préparation des données d'interaction entre les actions pour la formation

Si vous utilisez la recette personnalisée Next-Best-Action, Amazon Personalize utilise les données relatives aux interactions d'action pour identifier les intérêts des utilisateurs et prévoir les actions

les plus susceptibles d'être entreprises. Une interaction d'action est une interaction impliquant un utilisateur et une action dans votre jeu de données Actions. Par exemple, si vous avez une action d'inscription dans votre jeu de données Actions et qu'un utilisateur effectue cette action, vous devez enregistrer l'ID de l'utilisateur, l'ID de l'action, l'horodatage et, pour le type d'événement, enregistrer. `TAKEN`

Vous importez des interactions d'action dans un ensemble de données d'interactions Amazon Personalize Action. Vous pouvez importer des événements d'interaction d'action en masse avec une tâche d'importation de jeu de données, ou vous pouvez les diffuser en temps réel avec l'opération PutActionInteractions d'API. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine.

Les données relatives à vos interactions par actions groupées doivent se trouver dans un fichier CSV. Chaque ligne du fichier doit représenter une interaction unique entre un utilisateur et une action. Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Les sections suivantes fournissent plus d'informations sur la façon de préparer vos données d'interaction avec les actions pour Amazon Personalize. Pour les directives relatives au format des données en masse pour tous les types de données, consultez les directives relatives au format des données en masse.

Rubriques

- Exigences relatives aux données relatives aux interactions entre actions
- Données relatives au type d'événement
- Exemple de données sur les interactions entre actions

Exigences relatives aux données relatives aux interactions entre actions

Il n'y a aucune exigence minimale pour les données relatives aux interactions entre actions. Nous vous recommandons de l'importer pour des recommandations d'actions de qualité. Si vous ne disposez pas de données d'interaction d'action, vous pouvez créer un jeu de données d'interactions d'action vide et enregistrer les interactions de vos clients avec les actions à l'aide de l'opération d'PutActionInteractionsAPI.

Les données de vos interactions d'action doivent comporter au minimum les colonnes suivantes. Vous êtes libre d'ajouter des colonnes personnalisées supplémentaires en fonction de votre cas d'utilisation et de vos données.

- USER_ID — Identifiant unique de l'utilisateur qui a interagi avec l'élément. Chaque événement doit avoir un USER_ID. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- ACTION_ID — Identifiant unique de l'élément avec lequel l'utilisateur a interagi. Chaque événement doit avoir un identifiant d'article. Il doit s'agir `string` d'une longueur maximale de 256 caractères.
- TIMESTAMP — Heure à laquelle l'événement s'est produit (au format Unix Epoch en secondes). Chaque interaction d'action doit avoir un TIMESTAMP. Pour de plus amples informations, veuillez consulter Données d'horodatage.
- EVENT_TYPE — Indique si l'action a été entreprise, non effectuée ou visualisée. Chaque interaction d'action doit avoir un type d'événement. Pour de plus amples informations, veuillez consulter Données relatives au type d'événement.

Jusqu'à ce que vous importiez les données d'interaction entre les actions, Amazon Personalize recommande des actions dans votre boîte sans personnalisation, et les scores de propension sont de 0,0. Une action aura un score une fois qu'elle aura obtenu les résultats suivants :

- Au moins 50 interactions d'action avec le type d'événement TAKEN.
- Au moins 50 interactions d'action avec le type d'événement NOT_TAKEN ou VIEWED.

Ces interactions d'action doivent être présentes lors de la dernière formation sur la version de la solution et doivent avoir lieu dans un délai de 6 semaines à compter de l'horodatage de la dernière interaction dans le jeu de données d'interactions d'action.

Données relatives au type d'événement

Amazon Personalize peut utiliser des modèles dans les données relatives aux types d'événements pour identifier les actions les plus susceptibles d'être entreprises par vos utilisateurs. Par exemple, si un client ignore fréquemment une action d'abonnement par e-mail (indiquée par le type d'événement NOT_TAKEN), Amazon Personalize peut ajuster les recommandations afin de proposer moins de ce type d'action.

Vous ne pouvez utiliser que les types d'événements suivants pour les événements d'interaction entre actions. Amazon Personalize utilise ces événements pour en savoir plus sur votre utilisateur et calculer les prochaines actions à recommander.

- Taken : enregistre les événements Taken lorsqu'un utilisateur exécute une action recommandée.
- Non prise — Enregistrez les événements non effectués lorsque votre utilisateur fait le choix délibéré de ne pas effectuer l'action après l'avoir consultée. Par exemple, s'ils choisissent Non lorsque vous leur montrez l'action. Les événements Not Taken peuvent indiquer que le client n'est pas intéressé par l'action.
- Consulté — Enregistrez les événements consultés lorsque vous montrez une action à un utilisateur avant qu'il ne fasse le choix d'effectuer ou de ne pas effectuer une action. Amazon Personalize utilise les événements View pour en savoir plus sur les centres d'intérêt de vos utilisateurs. Par exemple, si un utilisateur voit une action mais ne l'exécute pas, il se peut que cette action ne l'intéresse pas à l'avenir.

Exemple de données sur les interactions entre actions

Les premières lignes d'un fichier CSV contenant les données d'interaction entre les actions et toutes les colonnes requises peuvent ressembler à ce qui suit.

```
USER_ID,ACTION_ID,EVENT_TYPE,TIMESTAMP
35,73,Viewed,1586731606
54,35,Not taken,1586731609
9,33,Viewed,1586735158
23,10,Taken,1586735697
27,11,Taken,1586735763
...
...
```

Une fois que vous avez terminé de préparer vos données, vous êtes prêt à créer un fichier de schéma JSON. Ce fichier indique à Amazon Personalize la structure de vos données. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize. Voici à quoi ressemblerait le fichier de schéma JSON pour les exemples de données ci-dessus.

```
{

  "type": "record",
```

```
  "name": "ActionInteractions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ACTION_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

Création de fichiers JSON pour les schémas Amazon Personalize

Après avoir préparé vos données, vous êtes prêt à créer des fichiers de schéma JSON pour chaque type de données que vous importez. Ces fichiers décrivent la structure et le contenu de vos données, y compris les noms des colonnes et leurs types de données.

Vous utilisez des fichiers JSON de schéma lorsque vous créez un schéma Amazon Personalize dansCréation d'un schéma et d'un jeu de données. Dans Amazon Personalize, un schéma est une ressource qui permet à Amazon Personalize d'analyser les données lorsque vous les importez dans votre ensemble de données. Vous créez un schéma pour chaque jeu de données que vous utilisez.

Pour les ressources personnalisées, chaque ensemble de données a des exigences de schéma spécifiques. Pour les groupes de jeux de données de domaine, le domaine que vous choisissez détermine les exigences de votre jeu de données et de schéma. Chaque domaine possède un schéma par défaut pour chaque type de jeu de données. Lorsque vous créez un jeu de données, vous pouvez utiliser le schéma de domaine existant ou en créer un nouveau en modifiant le schéma par défaut existant. Utilisez le schéma par défaut comme guide pour déterminer les données à importer pour votre domaine.

Les sections suivantes présentent les exigences personnalisées et relatives au domaine pour la création d'un fichier JSON de schéma pour chaque type de jeu de données.

Rubriques

- Exigences relatives au formatage du schéma
- Ensembles de données et schémas VIDEO_ON_DEMAND
- Ensembles de données et schémas de commerce électronique
- Ensembles de données et schémas personnalisés

Exigences relatives au formatage du schéma

Lorsque vous créez un schéma pour un ensemble de données dans un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé, vous devez suivre les directives suivantes :

- Vous devez définir le schéma au format Avro. Pour plus d'informations sur les types de données Avro que nous prenons en charge, consultezTypes de données de schéma.
- Un schéma possède une clé de nom dont la valeur doit correspondre au type de jeu de données.
- Les champs du schéma peuvent apparaître dans n'importe quel ordre, mais ils doivent correspondre à l'ordre des en-têtes de colonne correspondants dans votre fichier CSV.
- Les schémas doivent être des fichiers JSON plats sans structures imbriquées. Par exemple, un champ ne peut pas être le parent de plusieurs sous-champs.
- Les schémas Amazon Personalize ne prennent pas en charge les types complexes tels que les tableaux et les cartes.
- Les champs du schéma doivent avoir des noms alphanumériques uniques. Par exemple, vous ne pouvez pas ajouter à la fois un `GENRES_FIELD_1` champ et un `GENRESFIELD1` champ.
- Vous devez définir les champs obligatoires en tant que types de données obligatoires. Les champs de chaîne catégorielle réservés doivent avoir l'`categorical`attribut défini sur`true`, tandis que les champs de chaîne réservés ne peuvent pas être catégoriques. Les mots clés ne peuvent pas figurer dans vos données.
- Si vous ajoutez votre propre champ de métadonnées `string` et que vous souhaitez qu'Amazon Personalize l'utilise lors de la formation, il doit inclure l'`categorical`attribut ou l'`textual`attribut (seuls les schémas d'articles prennent en charge les champs comportant l'attribut textuel).
- Amazon Personalize n'utilise pas `boolean` de données de type lors de la formation ou du filtrage des recommandations. Pour qu'Amazon Personalize utilise des données booléennes lors de l'entraînement ou du filtrage, utilisez un champ de type String et utilisez les valeurs `"True"` et contenues `"False"` dans vos données. Vous pouvez également utiliser le type int ou long et les valeurs `0` et`1`.
- Les champs textuels doivent être du type `string` et leur `textual` attribut doit être défini sur. `true` Pour plus d'informations sur les données de texte non structurées, consultezMétadonnées de texte non structurées.

Les ensembles de données de groupes de jeux de données de domaine ont des exigences supplémentaires basées à la fois sur le domaine et le type de jeu de données. Les ensembles de données de groupes de jeux de données personnalisés ont des exigences supplémentaires en fonction du type.

Types de données de schéma

Les schémas Amazon Personalize prennent en charge les types Avro suivants pour les champs :

- float
- double
- int
- long
- chaîne
- un booléen
- null

Certains champs obligatoires et réservés prennent en charge les données nulles. L'ajout d'un `null` type à un champ vous permet d'utiliser des données imparfaites (par exemple, des métadonnées avec des valeurs vides) pour générer des recommandations. Pour plus d'informations sur les champs qui prennent en charge les données nulles, consultez la rubrique relative aux exigences du schéma pour votre domaine : Ensembles de données et schémas VIDEO_ON_DEMANDEnsembles de données et schémas de commerce électronique, ouEnsembles de données et schémas personnalisés. L'exemple suivant montre comment ajouter un type nul pour un champ GENDER.

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
}
```

Ensembles de données et schémas VIDEO_ON_DEMAND

Lorsque vous créez un groupe de jeux de données de domaine pour le domaine VIDEO_ON_DEMAND, chaque type de jeu de données possède un schéma par défaut avec un ensemble de champs obligatoires et recommandés spécifiques à VIDEO_ON_DEMAND. Vous pouvez utiliser le schéma par défaut ou en créer un nouveau basé sur le schéma par défaut. Les données que vous importez doivent correspondre au format et au type de votre schéma. Utilisez les schémas de domaine par défaut répertoriés dans les sections ci-dessous comme guide pour déterminer les données à importer afin de créer votre système de recommandation basé sur Video_on_Demand.

Vous êtes libre d'ajouter des champs supplémentaires. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Les rubriques suivantes fournissent des informations sur les champs obligatoires et recommandés de chaque ensemble de données pour le domaine VIDEO_ON_DEMAND. Chaque section du jeu de données inclut le schéma VIDEO_ON_DEMAND par défaut au format JSON.

Rubriques

- Exigences relatives au jeu de données et au schéma du domaine VIDEO_ON_DEMAND
- Exigences relatives au jeu de données sur les interactions entre articles (domaine VIDEO_ON_DEMAND)
- Exigences relatives aux ensembles de données des utilisateurs (domaine VIDEO_ON_DEMAND)
- Exigences relatives au jeu de données d'articles (domaine VIDEO_ON_DEMAND)

Exigences relatives au jeu de données et au schéma du domaine VIDEO_ON_DEMAND

Chaque type de jeu de données comporte les champs obligatoires et les mots clés réservés suivants. Les mots clés réservés sont des champs facultatifs qui ne sont pas des champs de métadonnées. Ces champs sont considérés comme réservés car vous devez définir les champs en tant que type de données obligatoire lorsque vous les utilisez. Les champs de chaîne catégorielle réservés doivent être `categorical` définis sur`true`, tandis que les champs de chaîne réservés ne peuvent pas être catégoriques. Les mots clés ne peuvent pas figurer dans vos données.

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
Interactions entre éléments (schéma par défaut)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`)	VALEUR_ÉVÉNEMENT (,) `float null` IMPRESSION (`string`,`null`)

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
	EVENT_TYPE (`string`et selon le cas d'utilisation `Watch` et les types d'`Click`événements)	IDENTIFIANT DE RECOMMANDATION (,) `string` `null` SOURCE D'ATTRIBUTION DE L'ÉVÉNEMENT (,) `string` `null`
Utilisateurs (schéma par défaut)	USER_ID (`string`) 1 champ de métadonnées (catégorique `string` ou numérique)	SUBSCRIPTION_MODEL (catégorique,) `string` `null`
Éléments (schéma par défaut)	ITEM_ID (`string`) HORODATAGE DE CRÉATION () `long` GENRES (catégori quesstring)	PRIX (`float`,`null`) DURÉE (`float`,`null`) GENRE_L2 (catégorique,) `string` `null` GENRE_L3 (catégorique,) `string` `null` NOTE MOYENNE (,) `float` `null` DESCRIPTION DU PRODUIT (`string`textuelle,) `null` CONTENT_OWNER (catégori questring,) `null` CONTENT_CLASSIFICATION (catégoriquestring,) `null`

Exigences relatives au jeu de données sur les interactions entre articles (domaine VIDEO_ON_DEMAND)

Un jeu de données sur les interactions entre les articles stocke les données historiques et en temps réel relatives aux interactions entre les utilisateurs et les articles de votre catalogue VIDEO_ON_DEMAND. Pour plus d'informations sur les types de données que vous pouvez stocker dans un jeu de données d'interactions, consultezDonnées d'interaction entre les articles.

Vous devez disposer d'un jeu de données d'interactions entre les éléments pour tous les cas d'utilisation et votre schéma doit comporter les champs suivants :

- USER_ID (`string`)
- IDENTIFIANT DE L'ARTICLE `string`
- TIMESTAMP (`long`)
- EVENT_TYPE (`string`et selon le cas d'utilisation `Watch` et les types d'`Click`événements)

Votre schéma peut également inclure les mots clés réservés suivants :

- VALEUR_ÉVÉNEMENT (,) `float` `null`
- IMPRESSION (`string`,`null`)
- IDENTIFIANT DE RECOMMANDATION (,) `string` `null`

Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous. Pour un exemple du schéma par défaut pour les ensembles de données d'interactions entre articles pour les domaines VIDEO_ON_DEMAND, consultez. Schéma d'interactions par défaut (domaine VIDEO_ON_DEMAND)

Ajoutez éventuellement le mot clé réservé EVENT_VALUE si vous disposez de données de valeur pour des événements, telles que le pourcentage d'une vidéo visionnée. Ajoutez éventuellement le mot clé réservé IMPRESSION si vous souhaitez inclure des données d'impressions explicites et implicites. Pour plus d'informations sur l'enregistrement des données d'impressions, voirDonnées sur les impressions.

Le nombre total maximum de champs de métadonnées facultatifs que vous pouvez ajouter à un jeu de données d'interactions entre articles, combiné au nombre total de types d'événements distincts

dans vos données d'interaction avec les articles, est de 10. Les champs de métadonnées inclus dans ce décompte sont les champs EVENT_TYPE, EVENT_VALUE ainsi que les champs de métadonnées personnalisés que vous ajoutez à votre schéma. Le nombre maximum de champs de métadonnées, à l'exception des champs réservés, tels que IMPRESSION, est de 5. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez une interaction avec une valeur catégorielle supérieure à 1 000, votre tâche d'importation de jeu de données échouera.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un jeu de données d'interactions entre articles pour le domaine VIDEO_ON_DEMAND, consultez. Quotas de service Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Schéma d'interactions par défaut (domaine VIDEO_ON_DEMAND)

Voici le schéma de domaine VIDEO_ON_DEMAND par défaut pour les ensembles de données d'interactions entre articles.

```
{

  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
```

```
  "version": "1.0"
}
```

Exigences relatives aux ensembles de données des utilisateurs (domaine VIDEO_ON_DEMAND)

Un jeu de données Utilisateurs stocke les métadonnées relatives à vos utilisateurs. Cela peut inclure des informations telles que l'âge, les intérêts, le sexe et le niveau de fidélité de chaque utilisateur. Pour plus d'informations sur les types de données utilisateur que vous pouvez importer dans Amazon Personalize, consultezMétadonnées utilisateur. Pour plus d'informations sur les exigences générales relatives au schéma Amazon Personalize, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Un ensemble de données Users est facultatif pour tous les cas d'utilisation de VIDEO_ON_DEMAND. Si vous avez des données utilisateur, nous vous recommandons d'en créer une pour obtenir les recommandations les plus pertinentes. Si vous créez un jeu de données Users, votre schéma doit inclure les champs suivants.

- USER_ID
- 1 champ de métadonnées (catégorique `string` ou numérique)

Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous. Pour un exemple du schéma par défaut pour les ensembles de données Users pour les domaines VIDEO_ON_DEMAND, consultez. Schéma des utilisateurs par défaut (domaine VIDEO_ON_DEMAND)

Un `SUBSCRIPTION_MODEL` champ est inclus dans le schéma par défaut. Ce champ est un mot clé réservé facultatif dont le type `string` doit être défini sur. `true` Pour obtenir les meilleures recommandations, nous vous recommandons de conserver ce champ dans votre schéma si vous avez des informations sur le modèle d'abonnement concernant chacun de vos utilisateurs dans vos données. Les données que vous importez doivent correspondre à votre schéma.

Utilisation de données catégoriques

Pour utiliser des données catégorielles, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles

dans votre fichier CSV en masse et dans vos importations d'enregistrements individuels. Pour les utilisateurs appartenant à plusieurs catégories, séparez chaque valeur à l'aide de la barre verticale « | ». Par exemple, pour un champ SUBSCRIPTION_MODEL, les données d'un utilisateur peuvent être student|monthly|discount.

Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si un utilisateur possède une valeur catégorielle de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Schéma des utilisateurs par défaut (domaine VIDEO_ON_DEMAND)

Voici le schéma de domaine VIDEO_ON_DEMAND par défaut pour les ensembles de données Users.

```
{

  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "SUBSCRIPTION_MODEL",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

Exigences relatives au jeu de données d'articles (domaine VIDEO_ON_DEMAND)

Un jeu de données d'articles stocke les métadonnées relatives à vos articles dans votre catalogue. Cela peut inclure des informations telles que le prix, le genre et la disponibilité de chaque article. Pour plus d'informations sur les types de données d'articles que vous pouvez importer dans Amazon Personalize, consultezMétadonnées de l'article. Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données

de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Un jeu de données Items est requis pour certains cas d'utilisation (voirCas d'utilisation de VIDEO_ON_DEMAND). Lorsque cela est facultatif, nous vous recommandons tout de même d'en créer un pour obtenir les recommandations les plus pertinentes. Si vous créez un jeu de données Items, votre schéma doit inclure les champs suivants :

- ITEM_ID
- GENRES (catégoriques`string`)
- CREATION_TIMESTAMP (au format Unix Epoch Time)

Votre schéma peut également inclure les mots clés réservés suivants. Chaque mot clé indique le type de données requis et indique s'il prend en charge les données nulles. L'ajout du type nul est facultatif.

- PRIX (flottant)
- DURÉE (flottant)
- GENRE_L2 (catégorique,) `string` `null`
- GENRE_L3 (catégorique,) `string` `null`
- NOTE MOYENNE (,) `float` `null`
- DESCRIPTION DU PRODUIT (`string`textuelle,) `null`
- CONTENT_OWNER (catégorique`string`,`null`) : société propriétaire de la vidéo. Par exemple, les valeurs peuvent être HBO, Paramount et NBC.
- CONTENT_CLASSIFICATION (catégorique`string`,`null`) : évaluation du contenu. Par exemple, les valeurs peuvent être G, PG, PG-13, R, NC-17 et rated.

Pour obtenir les meilleures recommandations, nous vous recommandons de conserver ces champs autant de champs dans votre schéma que vous disposez de données. Les données que vous importez doivent correspondre à votre schéma. Le nombre maximum de colonnes de métadonnées est de 100. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Utilisez les mots clés réservés GENRE_L2 et GENRE_L3 pour les articles comportant plusieurs catégories à plusieurs niveaux. Pour de plus amples informations, veuillez consulter Utilisation de

données catégoriques. Pour plus d'informations sur les métadonnées textuelles et catégoriques, voir. Préparation des métadonnées des éléments pour la formation Pour un exemple du schéma par défaut pour les ensembles de données Items pour les domaines de commerce électronique, consultezSchéma des articles par défaut (domaine VIDEO_ON_DEMAND).

Utilisation de données catégoriques

Pour utiliser des données catégorielles, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans les importations d'articles individuels. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez un élément avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Pour les articles comportant plusieurs catégories, séparez chaque valeur par la barre verticale « | ». Par exemple, pour un champ GENRES, vos données relatives à un article peuvent êtreAction|Crime|Biopic. Si vous disposez de plusieurs niveaux de données catégorielles et que certains éléments comportent plusieurs catégories pour chaque niveau de la hiérarchie, ajoutez un champ pour chaque niveau et ajoutez un indicateur de niveau après le nom de chaque champ : GENRES, GENRE_L2, GENRE_L3. Cela vous permet de filtrer les recommandations en fonction de sous-catégories, même si un article appartient à plusieurs catégories à plusieurs niveaux. Par exemple, une vidéo peut contenir les données suivantes pour chaque niveau de catégorie :

- GENRES : Action|Aventure
- GENRE_L2 : Crime | Western
- GENRE_L3 : film biographique

Dans cet exemple, la vidéo se trouve dans la hiérarchie action > crime > biopic et dans la hiérarchie aventure > western > biopic. Nous vous recommandons de n'utiliser que le niveau L3, mais vous pouvez utiliser d'autres niveaux si nécessaire. Pour plus d'informations sur la création et l'utilisation de filtres, consultezRecommandations de filtrage et segments d'utilisateurs.

Schéma des articles par défaut (domaine VIDEO_ON_DEMAND)

Le schéma par défaut pour les ensembles de données Items du domaine VIDEO_ON_DEMAND est le suivant.

```
{
  "type": "record",
```

```
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Ensembles de données et schémas de commerce électronique

Lorsque vous créez un groupe de jeux de données de domaine pour le domaine ECOMMERCE, chaque type de jeu de données possède un schéma par défaut avec un ensemble de champs obligatoires et recommandés spécifiques au commerce électronique. Vous pouvez utiliser le schéma par défaut ou en créer un nouveau basé sur le schéma par défaut. Les données que vous importez doivent correspondre au format et au type de votre schéma. Utilisez les schémas de domaine par défaut répertoriés dans les sections ci-dessous comme guide pour déterminer les données à importer afin de créer votre système de recommandation basé sur le commerce électronique.

Vous êtes libre d'ajouter des champs supplémentaires. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Les rubriques suivantes fournissent des informations sur les champs obligatoires et recommandés de chaque ensemble de données pour le domaine ECOMMERCE. Chaque section du jeu de données inclut le schéma ECOMMERCE par défaut au format JSON.

Rubriques

- Exigences relatives au jeu de données et au schéma du domaine ECOMMERCE
- Exigences relatives au jeu de données sur les interactions entre articles (domaine ECOMMERCE)
- Exigences relatives aux ensembles de données des utilisateurs (domaine ECOMMERCE)
- Exigences relatives au jeu de données sur les articles (domaine ECOMMERCE)

Exigences relatives au jeu de données et au schéma du domaine ECOMMERCE

Chaque type de jeu de données comporte les champs obligatoires et les mots clés réservés suivants. Les mots clés réservés sont des champs facultatifs qui ne sont pas des champs de métadonnées. Ces champs sont considérés comme réservés car vous devez définir les champs en tant que type de données obligatoire lorsque vous les utilisez. Les champs de chaîne catégorielle réservés doivent être `categorical` définis sur`true`, tandis que les champs de chaîne réservés ne peuvent pas être catégoriques. Les mots clés ne peuvent pas figurer dans vos données.

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
Interactions entre éléments (schéma par défaut)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`) EVENT_TYPE (`string`et selon le cas d'utilisation `Purchase` et les types d'`View`événements)	VALEUR_ÉVÉNEMENT (,) `float null` IMPRESSION (`string`,`null`) IDENTIFIANT DE RECOMMANDATION (,) `string null` SOURCE D'ATTRIBUTION DE L'ÉVÉNEMENT (,) `string null`
Utilisateurs (schéma par défaut)	USER_ID (`string`)	

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
	1 champ de métadonnées (catégorique `string` ou numérique)	
Eléments (schéma par défaut)	ITEM_ID (`string`) PRIX (`float`) CATEGORY_L1 (catégorique) `string`	CATEGORY_L2 (catégorique,) `string` `null` CATEGORY_L3 (catégorique,) `string` `null` DESCRIPTION DU PRODUIT (`string`textuelle,) `null` HORODATAGE DE CRÉATION () `long` AGE_GROUP (catégori questring,) `null` ADULTE (catégori questring,`null`) GENRE (catégori questring,`null`)

Exigences relatives au jeu de données sur les interactions entre articles (domaine ECOMMERCE)

Un jeu de données sur les interactions entre les articles stocke les données historiques et en temps réel relatives aux interactions entre les utilisateurs et les articles de votre catalogue de commerce électronique. Pour plus d'informations sur les types de données que vous pouvez stocker dans un jeu de données d'interactions, consultezDonnées d'interaction entre les articles. Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Vous devez au minimum créer un jeu de données d'interactions entre éléments et votre schéma doit comporter les champs suivants :

- USER_ID (`string`)
- ITEM_ID (`string`)
- TIMESTAMP (`long`)
- EVENT_TYPE (`string`et selon le cas d'utilisation `Purchase` et les types d'`View`événements)

Votre schéma peut également inclure les mots clés réservés suivants :

- VALEUR_ÉVÉNEMENT (,) `float` `null`
- IMPRESSION (`string`,`null`)
- IDENTIFIANT DE RECOMMANDATION (,) `string` `null`

Les données que vous importez doivent correspondre à votre schéma. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous. Pour un exemple du schéma par défaut pour les ensembles de données d'interactions entre articles pour les domaines de commerce électronique, consultezSchéma d'interactions par défaut (domaine ECOMMERCE).

Ajoutez éventuellement le mot clé réservé EVENT_VALUE si vous disposez de données de valeur pour les événements. Ajoutez éventuellement le mot clé réservé IMPRESSION si vous souhaitez inclure des données d'impressions explicites et implicites. Pour plus d'informations sur l'enregistrement des données d'impressions, voirDonnées sur les impressions.

Le nombre total maximum de champs de métadonnées facultatifs que vous pouvez ajouter à un jeu de données d'interactions entre articles, combiné au nombre total de types d'événements distincts dans vos données d'interaction avec les articles, est de 10. Les champs de métadonnées inclus dans ce décompte sont les champs EVENT_TYPE, EVENT_VALUE ainsi que les champs de métadonnées personnalisés que vous ajoutez à votre schéma. Le nombre maximum de champs de métadonnées, à l'exception des champs réservés, tels que IMPRESSION, est de 5. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez une interaction avec une valeur catégorielle supérieure à 1 000, votre tâche d'importation de jeu de données échouera.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un ensemble de données d'interactions entre articles pour le domaine ECOMMERCE, voirQuotas de service.

Schéma d'interactions par défaut (domaine ECOMMERCE)

Voici le schéma de domaine ECOMMERCE par défaut pour les ensembles de données d'interactions entre articles.

```
{

  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

Exigences relatives aux ensembles de données des utilisateurs (domaine ECOMMERCE)

Un jeu de données Utilisateurs stocke les métadonnées relatives à vos utilisateurs. Cela peut inclure des informations telles que l'âge, le sexe et le niveau de fidélité de chaque utilisateur. Pour plus d'informations sur les types de données utilisateur que vous pouvez importer dans Amazon

Personalize, consultezMétadonnées utilisateur. Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Un ensemble de données utilisateurs est facultatif pour tous les cas d'utilisation du commerce électronique. Si vous avez des données utilisateur, nous vous recommandons d'en créer une pour obtenir les recommandations les plus pertinentes. Si vous créez un jeu de données Users, votre schéma doit inclure les champs suivants.

- USER_ID
- 1 champ de métadonnées (catégorique `string` ou numérique)

Les données que vous importez doivent correspondre à votre schéma. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous. Pour un exemple du schéma par défaut pour les ensembles de données Utilisateurs pour les domaines de commerce électronique, voirSchéma des utilisateurs par défaut (domaine ECOMMERCE).

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un ensemble de données utilisateurs, consultezQuotas de service.

Utilisation de données catégoriques

Pour utiliser des données catégoriques, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans vos importations d'enregistrements individuels. Pour les utilisateurs ayant plusieurs catégories, séparez chaque valeur à l'aide de la barre verticale « | ». Par exemple, pour un champ SUBSCRIPTION_MODEL, les données d'un utilisateur peuvent être student|monthly|discount.

Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si un utilisateur possède une valeur catégorielle de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Schéma des utilisateurs par défaut (domaine ECOMMERCE)

Voici le schéma de domaine ECOMMERCE par défaut pour les ensembles de données des utilisateurs avec un champ CATEGORY comme champ de métadonnées obligatoire.

```
{

  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "MEMBERSHIP_STATUS",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

Exigences relatives au jeu de données sur les articles (domaine ECOMMERCE)

Un jeu de données d'articles stocke les métadonnées relatives à vos articles de commerce électronique. Cela peut inclure des informations telles que le prix, la catégorie et la description du produit pour chaque article. Pour plus d'informations sur les types de données d'articles que vous pouvez importer dans Amazon Personalize, consultezMétadonnées de l'article. Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas, quel que soit le domaine.

Un ensemble de données d'articles est facultatif pour tous les cas d'utilisation du commerce électronique. Si vous avez des données sur les articles, nous vous recommandons d'en créer une pour obtenir les recommandations les plus pertinentes. Si vous créez un jeu de données d'éléments, votre schéma doit inclure les champs suivants :

- ITEM_ID
- PRIX (`float`)
- CATEGORY_L1 (catégorique) `string`

Votre schéma peut également inclure les mots clés réservés suivants. Pour les champs catégoriels, vous pouvez définir votre propre plage de valeurs en fonction de votre cas d'utilisation.

- CATEGORY_L2 (catégorique,) `string` `null`
- CATEGORY_L3 (catégorique,) `string` `null`
- DESCRIPTION DU PRODUIT (`string`textuelle,) `null`
- HORODATAGE DE CRÉATION () `float`
- AGE_GROUP (catégorique`string`,`null`) : groupe d'âge auquel l'article est destiné. Les valeurs peuvent être les nouveau-nés, les nourrissons, les enfants et les adultes.
- ADULTE (catégorique`string`,`null`) : Indique si l'article est réservé aux adultes, comme l'alcool. Les valeurs peuvent être « oui » ou « non ».
- GENRE (catégorique`string`,`null`) : Le sexe auquel l'article est destiné. Les valeurs peuvent être masculines, féminines ou unisexes.

Pour obtenir les meilleures recommandations, nous vous recommandons de conserver ces champs autant de champs dans votre schéma que vous disposez de données. Les données que vous importez doivent correspondre à votre schéma. Les données que vous importez doivent correspondre à votre schéma. Le nombre maximum de colonnes de métadonnées est de 100. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Utilisez les mots clés réservés CATEGORY_L2 et CATEGORY_L3 pour les articles comportant plusieurs catégories à plusieurs niveaux. Pour de plus amples informations, veuillez consulter Utilisation de données catégoriques. Pour plus d'informations sur les métadonnées textuelles et catégoriques, voir. Métadonnées de texte non structurées Pour un exemple du schéma par défaut pour les ensembles de données Items pour les domaines de commerce électronique, consultezSchéma des articles par défaut (domaine ECOMMERCE).

Utilisation de données catégoriques

Pour utiliser des données catégoriques, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans les importations d'articles individuels. Vous pouvez définir votre propre plage de valeurs en fonction de votre cas d'utilisation. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez un élément avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Pour les articles comportant plusieurs catégories, séparez chaque valeur par la barre verticale « | ». Par exemple, pour un champ CATEGORY_L1, vos données pour un élément peuvent être. `Electronics|Productivity|Mouse` Si vous disposez de plusieurs niveaux de données catégorielles et que certains éléments comportent plusieurs catégories pour chaque niveau de la hiérarchie, ajoutez un champ pour chaque niveau et ajoutez un indicateur de niveau après le nom de chaque champ : CATEGORY_L1, CATEGORY_L2, CATEGORY_L3. Cela vous permet de filtrer les recommandations en fonction de sous-catégories, même si un article appartient à plusieurs catégories à plusieurs niveaux. Par exemple, un article peut contenir les données suivantes pour chaque niveau de catégorie :

- CATEGORIE_L1 : Électronique|Productivité
- CATÉGORIE L2 : Productivité|Ordinateurs
- CATEGORY_L3 : Souris

Dans cet exemple, l'élément se trouve dans la hiérarchie électronique > productivité > souris et dans la hiérarchie productivité > ordinateurs > souris. Nous vous recommandons de n'utiliser que le niveau L3, mais vous pouvez utiliser d'autres niveaux si nécessaire. Pour plus d'informations sur la création et l'utilisation de filtres, voirRecommandations de filtrage et segments d'utilisateurs.

Schéma des articles par défaut (domaine ECOMMERCE)

Voici le schéma par défaut pour les ensembles de données Items pour le domaine ECOMMERCE avec uniquement les champs obligatoires.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
```

```
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "PRICE",
      "type": "float"
    },
    {
      "name": "CATEGORY_L1",
      "type": [
        "string"
      ],
      "categorical": true
    }
  ],
  "version": "1.0"
}
```

Ensembles de données et schémas personnalisés

Lorsque vous créez un groupe de jeux de données personnalisé, vous créez vos propres schémas à partir de zéro. Les ensembles de données et les schémas de groupes de jeux de données personnalisés comportent moins de champs obligatoires et offrent une plus grande flexibilité. Les rubriques suivantes expliquent le schéma et les exigences en matière de données pour les ensembles de données d'un groupe de jeux de données personnalisé. Chaque section du jeu de données répertorie les données requises pour le type d'ensemble de données et fournit un exemple de schéma JSON.

Pour plus d'informations sur les types de données que vous pouvez importer dans Amazon Personalize, consultezTypes de données qu'Amazon Personalize peut utiliser. Pour plus d'informations sur les exigences générales du schéma Amazon Personalize, telles que les exigences de mise en forme et les types de données de champ disponibles, consultezCréation de fichiers JSON pour les schémas Amazon Personalize. Ces exigences s'appliquent à tous les schémas Amazon Personalize.

Rubriques

- Exigences relatives aux ensembles de données et aux schémas personnalisés
- Exigences relatives au schéma du jeu de données sur les interactions entre articles (personnalisé)
- Exigences relatives au schéma du jeu de données utilisateur (personnalisé)

- Exigences relatives au schéma du jeu de données d'articles (personnalisé)
- Exigences relatives au schéma du jeu de données Actions (personnalisé)
- Exigences relatives au schéma du jeu de données Action Interactions (personnalisé)

Exigences relatives aux ensembles de données et aux schémas personnalisés

Lorsque vous créez un ensemble de données pour un groupe de jeux de données personnalisé, chaque type de jeu de données comporte les champs obligatoires et les mots clés réservés suivants avec les types de données requis.

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
Interactions entre éléments (exemple de schéma)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`)	EVENT_TYPE (`string`) VALEUR_ÉVÉNEMENT (,) `float null` IMPRESSION (`string,null`) IDENTIFIANT DE RECOMMANDATION (,) `string null` SOURCE D'ATTRIBUTION DE L'ÉVÉNEMENT (,) `string null`
Utilisateurs (exemple de schéma)	USER_ID (`string`) 1 champ de métadonnées (catégorique `string` ou numérique)	
Éléments (exemple de schéma)	ITEM_ID (`string`)	HORODATAGE DE CRÉATION () `long`

Type d'ensemble de données	Champs obligatoires	Mots-clés réservés
	1 champ de métadonnées (champ catégorique ou textuel ou `string` champ numérique)	
Actions (exemple de schéma)	IDENTIFIANT DE L'ACTION () `string` 1 champ de métadonnées (catégorique `string` ou numérique)	HORODATAGE DE CRÉATION () `long` VALEUR (`long`,`null`) TAPEZ (`string`,`null`) HORODATAGE D'EXPIRATION (,) `long null` FRÉQUENCE DE RÉPÉTITION (,) `long null`
Interactions entre les actions (exemple de schéma)	USER_ID (`string`) IDENTIFIANT DE L'ACTION () `string` EVENT_TYPE (`string`) TIMESTAMP (`long`)	IMPRESSION (`string`,`null`) IDENTIFIANT DE RECOMMANDATION (,) `string null`

Champs de métadonnées

Les métadonnées incluent des champs sous forme de chaîne ou non sous forme de chaîne qui ne sont pas obligatoires ou qui n'utilisent pas de mot clé réservé. Les schémas de métadonnées sont soumis aux restrictions suivantes :

- Les schémas d'utilisateurs, d'éléments et d'actions nécessitent au moins un champ de métadonnées.
- Vous pouvez ajouter au maximum 25 champs de métadonnées pour un schéma Users, 100 champs de métadonnées pour un schéma Items et 10 champs de métadonnées pour un schéma Actions.

- Si vous ajoutez votre propre champ de métadonnées de type`string`, il doit inclure l'`categorical`attribut ou l'`textual`attribut (seuls les schémas Items prennent en charge les champs avec l'attribut textuel). Sinon, Amazon Personalize n'utilisera pas le champ lors de la formation d'un modèle.

Mots-clés réservés

Les mots clés réservés sont des champs facultatifs qui ne sont pas des champs de métadonnées. Ces champs sont considérés comme réservés car vous devez définir les champs comme leur type de données obligatoire lorsque vous les utilisez, et les mots clés ne peuvent pas être utilisés comme valeurs dans vos données. Les champs de chaîne catégorielle réservés doivent être `categorical` définis sur`true`, tandis que les champs de chaîne réservés ne peuvent pas être catégoriques. Les mots clés suivants sont réservés :

- EVENT_TYPE : pour les ensembles de données d'interactions entre éléments comportant un ou plusieurs types d'événements, tels que le clic et le téléchargement, utilisez un champ. `EVENT_TYPE` Vous devez définir un champ EVENT_TYPE comme un champ `string` et ne pouvez pas être défini comme catégorique.
- EVENT_VALUE : pour les ensembles de données d'interactions entre éléments qui incluent des données de valeur relatives à des événements, telles que le pourcentage d'une vidéo regardée par un utilisateur, utilisez un `EVENT_VALUE` champ avec un type `float` et éventuellement. `null`
- CREATION_TIMESTAMP : pour les ensembles de données d'éléments ou d'actions avec un horodatage pour la date de création de chaque élément, utilisez un champ avec un type. `CREATION_TIMESTAMP` `long` Amazon Personalize utilise `CREATION_TIMESTAMP` les données pour calculer l'âge d'un article et ajuster les recommandations en conséquence. Consultez Données d'horodatage de création.
- IMPRESSION : Pour les ensembles de données d'interactions entre articles contenant des données d'impressions explicites, utilisez un `IMPRESSION` champ avec un type `String` et éventuellement un type`null`. Les impressions sont des listes d'éléments visibles par un utilisateur lorsqu'il interagissait avec un élément en particulier (par exemple, en cliquant ou en regardant). Pour de plus amples informations, veuillez consulter Données sur les impressions.
- RECOMMENDATION_ID : pour les ensembles de données d'interactions entre articles qui utilisent les recommandations précédentes comme données d'impressions implicites, utilisez éventuellement un `RECOMMENDATION_ID` champ avec type et éventuellement type`String`. `null`

Il n'est pas nécessaire d'ajouter un `RECOMMENDATION_ID` champ pour qu'Amazon Personalize utilise les impressions implicites lors de la génération de recommandations. Vous pouvez réussir `recommendationId` une PutEvents opération sans elle. Pour de plus amples informations, veuillez consulter Données sur les impressions.

- VALEUR : pour les ensembles de données Actions, si vous disposez de données de valeur pour certaines ou toutes vos actions, ajoutez un `VALUE` champ à votre schéma. Pour son type, utilisez `long` et tapez éventuellement`null`. Pour plus d'informations sur les actions et leur valeur, consultezDonnées de valeur.
- ACTION_EXPIRATION_TIMESTAMP : pour les ensembles de données Actions, si vous avez un horodatage d'expiration pour certaines ou toutes vos actions, ajoutez un champ à votre schéma. `ACTION_EXPIRATION_TIMESTAMP` Pour son type, utilisez `long` et tapez éventuellement`null`. Pour plus d'informations sur les horodatages d'expiration, consultez. Données d'horodatage d'expiration des actions
- REPEAT_FREQUENCY : pour les ensembles de données Actions, si vous disposez de données de fréquence de répétition pour certaines ou toutes vos actions, ajoutez un `REPEAT_FREQUENCY` champ à votre schéma. Pour son type, utilisez `long` et tapez éventuellement`null`. Pour plus d'informations sur les données de fréquence de répétition, consultezRépéter les données de fréquence.

Exigences relatives au schéma du jeu de données sur les interactions entre articles (personnalisé)

Un jeu de données sur les interactions entre les articles stocke les données historiques et en temps réel relatives aux interactions entre les utilisateurs et les articles de votre catalogue. Pour plus d'informations sur les types de données d'interactions qu'Amazon Personalize peut utiliser, consultezDonnées d'interaction entre les articles.

Les données que vous fournissez pour chaque interaction doivent correspondre à votre schéma. Selon votre schéma, les métadonnées d'interaction peuvent inclure des valeurs vide/nulles. Vous devez au minimum fournir les informations suivantes pour chaque interaction :

- ID de l’utilisateur
- Identifiant de l'article
- Horodatage (au format Unix Epoch Time)

Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Le nombre total maximum de champs de métadonnées facultatifs que vous pouvez ajouter à un jeu de données d'interactions entre articles, combiné au nombre total de types d'événements distincts dans vos données d'interaction entre articles, est de 10. Les champs de métadonnées inclus dans ce décompte sont les champs EVENT_TYPE, EVENT_VALUE ainsi que les champs de métadonnées personnalisés que vous ajoutez à votre schéma. Le nombre maximum de champs de métadonnées, à l'exception des champs réservés, tels que IMPRESSION, est de 5. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez une interaction avec une valeur catégorielle supérieure à 1 000, votre tâche d'importation de jeu de données échouera.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un jeu de données d'interactions entre éléments, consultezQuotas de service.

Exemple de schéma d'interactions (personnalisé)

L'exemple suivant montre le schéma d'un jeu de données d'interactions entre articles. Les champs obligatoires sont `USER_ID`, `ITEM_ID` et `TIMESTAMP`. Les `IMPRESSION` champs `EVENT_TYPEEVENT_VALUE`, et sont des mots clés réservés facultatifs reconnus par Amazon Personalize. EVENT_TYPE doit être de type chaîne et ne peut pas être catégorique. `LOCATIONet` `DEVICE` sont des champs de métadonnées contextuels facultatifs. Pour plus d'informations sur les exigences du schéma, voirExigences relatives aux ensembles de données et aux schémas personnalisés.

```
{

  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
```

```
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "EVENT_VALUE",
          "type": [
             "float",
             "null"
          ]
      },
      {
          "name": "LOCATION",
          "type": "string",
          "categorical": true
      },
      {
          "name": "DEVICE",
          "type": [
              "string",
              "null"
          ],
          "categorical": true
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      },
      {
          "name": "IMPRESSION",
          "type": "string"
      }
  ],
  "version": "1.0"
}
```

Pour ce schéma, les premières lignes de données historiques d'un fichier CSV peuvent ressembler à ce qui suit. Notez que certaines valeurs de EVENT_VALUE sont nulles.

```
USER_ID,ITEM_ID,EVENT_TYPE,EVENT_VALUE,LOCATION,DEVICE,TIMESTAMP,IMPRESSION
35,73,click,,Ohio,Tablet,1586731606,73|70|17|95|96|92|55|45|16|97|56|54|33|94|36|10|5|
43|19|13|51|90|65|59|38
```

```
54,35,watch,0.75,Indiana,Cellphone,1586735164,35|82|78|57|20|63|1|90|76|75|49|71|26|24|
25|6|37|85|40|98|32|13|11|54|48
9,33,click,,Oregon,Cellphone,1586735158,68|33|62|6|15|57|45|24|78|89|90|40|26|91|66|31|
47|17|99|29|27|41|77|75|14
23,10,watch,0.25,California,Tablet,1586735697,92|89|36|10|39|77|4|27|79|18|83|16|28|68|
78|40|50|3|99|7|87|49|12|57|53
27,11,watch,0.55,Indiana,Tablet,1586735763,11|7|39|95|71|1|6|40|41|28|99|53|68|76|0|65|
69|36|22|42|34|67|24|20|66
...
...
```

Exigences relatives au schéma du jeu de données utilisateur (personnalisé)

Un jeu de données Utilisateurs stocke les métadonnées relatives à vos utilisateurs. Cela peut inclure des informations telles que l'âge, le sexe et l'adhésion au programme de fidélité pour chaque article. Pour plus d'informations sur les types de données utilisateur que vous pouvez importer dans Amazon Personalize, consultezMétadonnées utilisateur.

Les données que vous fournissez pour chaque utilisateur doivent correspondre à votre schéma. Au minimum, vous devez fournir un nom d'utilisateur pour chaque utilisateur (256 caractères maximum). Selon votre schéma, les métadonnées utilisateur peuvent inclure des empty/null valeurs. Votre schéma Users doit comporter au moins un champ de métadonnées, mais si vous ajoutez un `null` type, cette valeur peut être nulle pour l'utilisateur. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Pour utiliser des données catégoriques, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans vos importations d'enregistrements individuels. Pour les utilisateurs appartenant à plusieurs catégories, séparez chaque valeur à l'aide de la barre verticale « | ». Par exemple, pour un champ SUBSCRIPTION_MODEL, les données d'un utilisateur peuvent être student|monthly|discount.

Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si un utilisateur possède une valeur catégorielle de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un ensemble de données utilisateurs, consultezQuotas de service.

Exemple de schéma d'utilisateurs (personnalisé)

L'exemple suivant montre comment structurer un schéma Users. Le `USER_ID` champ est obligatoire et les `GENDER` champs `AGE` et sont des métadonnées. Au moins un champ de métadonnées est obligatoire et vous pouvez ajouter au maximum 25 champs de métadonnées. Pour plus d'informations sur les exigences du schéma, voirExigences relatives aux ensembles de données et aux schémas personnalisés.

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "AGE",
          "type": "int"
      },
      {
          "name": "GENDER",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

Pour ce schéma, les premières lignes de données historiques d'un fichier CSV peuvent ressembler à ce qui suit.

```
USER_ID,AGE,GENDER
5,34,Male
6,56,Female
8,65,Male
...
```

```
...
```

Exigences relatives au schéma du jeu de données d'articles (personnalisé)

Un jeu de données d'articles stocke les métadonnées relatives à vos articles dans votre catalogue. Cela peut inclure des informations telles que le prix, le genre et la disponibilité de chaque article. Pour plus d'informations sur les types de données d'articles que vous pouvez importer dans Amazon Personalize, consultezMétadonnées de l'article.

Les données que vous fournissez pour chaque élément doivent correspondre au schéma de votre jeu de données Items. Au minimum, vous devez fournir un identifiant d'article pour chaque article (longueur maximale de 256 caractères). Selon votre schéma, les métadonnées des éléments peuvent inclure des empty/null valeurs. Votre schéma doit comporter au moins un champ de métadonnées, mais si vous ajoutez un `null` type, cette valeur peut être nulle pour l'élément. Vous êtes libre d'ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Tant que les champs ne sont pas répertoriés comme obligatoires ou réservés et que les types de données y sont répertoriésTypes de données de schéma, les noms des champs et les types de données dépendent de vous.

Pour utiliser des données catégoriques, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans les importations d'articles individuels. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez un élément avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Pour les articles comportant plusieurs catégories, séparez chaque valeur par la barre verticale « | ». Par exemple, pour un champ GENRES, vos données relatives à un article peuvent être`Action|Crime|Biopic`. Si vous disposez de plusieurs niveaux de données catégorielles et que certains éléments comportent plusieurs catégories pour chaque niveau de la hiérarchie, ajoutez un champ pour chaque niveau et ajoutez un indicateur de niveau après le nom de chaque champ : GENRES, GENRE_L2, GENRE_L3. Cela vous permet de filtrer les recommandations en fonction de sous-catégories, même si un article appartient à plusieurs catégories à plusieurs niveaux (pour plus d'informations sur la création et l'utilisation de filtres, voirRecommandations de filtrage et segments d'utilisateurs). Par exemple, une vidéo peut contenir les données suivantes pour chaque niveau de catégorie :

- GENRES : Action|Aventure
- GENRE_L2 : Criminel|Western

- GENRE_L3 : Biopic

Dans cet exemple, la vidéo se trouve dans la hiérarchie action > crime > biopic et dans la hiérarchie aventure > western > biopic. Nous vous recommandons de n'utiliser que le niveau L3, mais vous pouvez utiliser d'autres niveaux si nécessaire.

Au cours de la formation des modèles, Amazon Personalize prend en compte un maximum de 750 000 articles. Si vous importez plus de 750 000 articles, Amazon Personalize décide quels articles inclure dans la formation, en mettant l'accent sur les nouveaux articles (articles que vous avez récemment ajoutés sans interaction) et les articles existants avec des données d'interactions récentes.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un jeu de données Items, consultezQuotas de service.

Exemple de schéma de jeu de données d'articles (personnalisé)

L'exemple suivant montre comment structurer un schéma Items. Le champ `ITEM_ID` est obligatoire. Le `GENRE` champ est constitué de métadonnées catégoriques et le `DESCRIPTION` champ de métadonnées textuelles. Au moins un champ de métadonnées est obligatoire. Vous pouvez ajouter un maximum de 100 champs de métadonnées. Le `CREATION_TIMESTAMP` champ est un mot clé réservé. Pour plus d'informations sur les exigences relatives aux schémas, consultezExigences relatives aux ensembles de données et aux schémas personnalisés.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
```

```
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "DESCRIPTION",
      "type": [
        "null",
        "string"
      ],
      "textual": true
    }
  ],
  "version": "1.0"
}
```

Pour ce schéma, les premières lignes de données historiques d'un fichier CSV peuvent ressembler à ce qui suit.

```
ITEM_ID,GENRES,CREATION_TIMESTAMP,DESCRIPTION
1,Adventure|Animation|Children|Comedy|Fantasy,1570003267,"This is an animated movie
 that features action, comedy, and fantasy. Audience is children. This movie was
 released in 2004."
2,Adventure|Children|Fantasy,1571730101,"This is an adventure movie with elements of
 fantasy. Audience is children. This movie was release in 2010."
3,Comedy|Romance,1560515629,"This is a romantic comedy. The movie was released in 1999.
 Audience is young women."
4,Comedy|Drama|Romance,1581670067,"This movie includes elements of both comedy and
 drama as well as romance. This movie was released in 2020."
...
...
```

Exigences relatives au schéma du jeu de données Actions (personnalisé)

Une action est une activité d'engagement que vous souhaiterez peut-être recommander à vos clients. Les actions peuvent inclure l'installation de votre application mobile, la création d'un profil de membre, l'adhésion à votre programme de fidélité ou l'inscription à des e-mails promotionnels. Un jeu de données Actions stocke les données relatives à vos actions. Pour plus d'informations sur les types de données d'action que vous pouvez importer dans Amazon Personalize, consultez[Métadonnées relatives aux actions](#).

Les données que vous fournissez pour chaque action doivent correspondre au schéma de votre jeu de données Actions. Selon votre schéma, les métadonnées des actions peuvent inclure des empty/null valeurs.

Au minimum, vous devez fournir un ID d'action pour chaque élément (longueur maximale de 256 caractères). Votre schéma doit comporter au moins un champ de métadonnées, mais si vous ajoutez un `null` type, cette valeur peut être nulle pour l'action. Vous pouvez ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Vous pouvez choisir les noms des champs et les types de données à moins que les champs ne soient répertoriés comme obligatoires ou réservés, et que les types de données ne soient répertoriés dansTypes de données de schéma.

Pour ajouter un champ catégorique, ajoutez un champ de type `string` et définissez l'attribut catégorique du champ sur `true` dans votre schéma. Incluez ensuite les données catégorielles dans votre fichier CSV en masse et dans les importations d'actions individuelles. Les valeurs catégorielles peuvent comporter au maximum 1 000 caractères. Si vous avez une action avec une valeur catégorique de plus de 1 000 caractères, votre tâche d'importation de jeu de données échouera.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un jeu de données Actions, consultezQuotas de service.

Exemple de schéma de jeu de données Actions (personnalisé)

L'exemple suivant montre comment structurer un schéma Actions. Le champ `ACTION_ID` est obligatoire. Le `MEMBERSHIP_LEVEL` champ est un champ de chaîne catégorique. Les `REPEAT_FREQUENCY` champs `VALUECREATION_TIMESTAMP`, et sont des mots clés réservés avec les types requis. Vous pouvez ajouter un maximum de 10 colonnes. Pour plus d'informations sur les exigences du schéma, consultezExigences relatives aux ensembles de données et aux schémas personnalisés.

```
{
  "type": "record",
  "name": "Actions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ACTION_ID",
      "type": "string"
```

```
    },
    {
      "name": "VALUE",
      "type": [
        "null",
        "long"
      ]
    },

    {
      "name": "MEMBERSHIP_LEVEL",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
    },

    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "REPEAT_FREQUENCY",
      "type": [
        "long",
        "null"
      ]
    }
  ],
  "version": "1.0"
}
```

Pour ce schéma, les premières lignes de données historiques d'un fichier CSV peuvent ressembler à ce qui suit.

```
ACTION_ID,VALUE,MEMBERSHIP_LEVEL,CREATION_TIMESTAMP,REPEAT_FREQUENCY
1,10,Deluxe|Premium,1510003267,7
2,5,Basic,1580003267,7
3,5,Preview,1590003267,3
4,10,Deluxe|Platinum,1560003267,4
...
...
```

Exigences relatives au schéma du jeu de données Action Interactions (personnalisé)

Un jeu de données d'interactions d'actions stocke des données historiques et en temps réel relatives aux interactions entre les utilisateurs et aux actions de votre jeu de données d'actions. Pour plus d'informations sur les types de données qu'Amazon Personalize peut utiliser, consultezDonnées d'interaction entre actions.

Les données que vous fournissez pour chaque interaction doivent correspondre à votre schéma. Selon votre schéma, les métadonnées d'interaction peuvent inclure des empty/null valeurs. Au minimum, votre schéma doit inclure les éléments suivants :

- USER_ID
- IDENTIFIANT_ACTION
- TIMESTAMP
- EVENT_TYPE

Vous pouvez ajouter des champs supplémentaires en fonction de votre cas d'utilisation et de vos données. Vous pouvez choisir les noms des champs et les types de données à moins que les champs ne soient répertoriés comme obligatoires ou réservés, et que les types de données ne soient répertoriés dansTypes de données de schéma.

Pour plus d'informations sur les exigences minimales et les limites de données maximales pour un jeu de données d'interactions Action, consultezQuotas de service.

Exemple de schéma de jeu de données d'interactions avec action (personnalisé)

L'exemple suivant montre le schéma d'un jeu de données d'interactions Action avec uniquement les champs obligatoires. Pour plus d'informations sur les exigences générales de mise en forme des schémas, consultezExigences relatives au formatage du schéma.

```
{

  "type": "record",
  "name": "ActionInteractions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
```

```
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ACTION_ID",
            "type": "string"
        },
        {
            "name": "EVENT_TYPE",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ],
    "version": "1.0"
}
```

Pour ce schéma, les premières lignes de données historiques d'un fichier CSV peuvent ressembler à ce qui suit.

```
USER_ID,ACTION_ID,EVENT_TYPE,TIMESTAMP
35,73,Viewed,1586731606
54,35,Not taken,1586731609
9,33,Viewed,1586735158
23,10,Taken,1586735697
27,11,Taken,1586735763
...
...
```

Liste de vérification du niveau de préparation

Cette liste de contrôle fournit des listes des fonctionnalités, des exigences et des instructions relatives aux données d'Amazon Personalize. Il peut vous aider à planifier, ou vous pouvez l'utiliser comme référence lorsque vous créez des ressources dans Amazon Personalize.

Rubriques

- Avez-vous adapté vos cas d'utilisation aux ressources Amazon Personalize ?
- Disposez-vous de suffisamment de données sur les interactions entre les articles ?
- Disposez-vous d'une architecture de diffusion d'événements en temps réel ?
- Vos données sont-elles optimisées pour Amazon Personalize ?
- Collectez-vous des données facultatives susceptibles d'améliorer les recommandations ?
- Avez-vous un plan pour tester vos recommandations ?
- Avez-vous d'autres objectifs commerciaux ?

Avez-vous adapté vos cas d'utilisation aux ressources Amazon Personalize ?

Les recommandations d'Amazon Personalize peuvent répondre aux cas d'utilisation suivants :

- Génération de recommandations personnalisées pour un utilisateur
- Recommander des articles similaires ou connexes
- Recommander des articles tendance ou populaires
- Recommander les meilleures actions suivantes à un utilisateur
- Réorganisation par pertinence (uniquement avec des ressources personnalisées)
- Génération de segments d'utilisateurs (uniquement avec des ressources personnalisées)

Amazon Personalize propose des ressources basées sur le domaine et des ressources personnalisées configurées pour ces cas d'utilisation. Vous commencez par créer un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé :

- Avec un groupe de jeux de données de domaine, vous créez des ressources préconfigurées et optimisées pour les domaines VIDEO_ON_DEMAND ou ECOMMERCE.

Si vous possédez une application de streaming vidéo ou de commerce électronique, nous vous recommandons de commencer par un groupe de jeux de données de domaine. Vous pouvez toujours ajouter des ressources personnalisées, telles que des solutions et des versions de solutions conçues pour des cas d'utilisation personnalisés. Et vous pouvez toujours utiliser des ressources personnalisées pour obtenir des recommandations par lots. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine.

- Avec un groupe de jeux de données personnalisé, vous choisissez une recette qui correspond à votre cas d'utilisation. Vous formez et déployez ensuite uniquement des solutions et des versions de solutions configurables (modèles de recommandation Amazon Personalize entraînés). Lorsque vous êtes prêt, vous pouvez déployer la version de la solution dans le cadre d'une campagne pour obtenir des recommandations en temps réel. Vous pouvez également obtenir des recommandations par lots sans campagne.

  Si vous n'avez pas d'application de streaming vidéo ou de commerce électronique, nous vous recommandons de créer un groupe de jeux de données personnalisé. Sinon, commencez par un groupe de jeux de données de domaine et ajoutez des ressources personnalisées si nécessaire.

Pour plus d'informations sur les cas d'utilisation et les recettes personnalisées disponibles dans Amazon Personalize, consultezAdapter votre cas d'utilisation aux ressources Amazon Personalize.

Disposez-vous de suffisamment de données sur les interactions entre les articles ?

Pour tous les cas d'utilisation et toutes les recettes, vous devez avoir au moins 1 000 interactions avec des objets pour 25 utilisateurs uniques avec au moins deux interactions chacun. Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

Si vous n'êtes pas sûr de disposer de suffisamment de données, vous pouvez les importer et les analyser à l'aide de la console Amazon Personalize. Pour de plus amples informations, veuillez consulter Analyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

Disposez-vous d'une architecture de diffusion d'événements en temps réel ?

Si vous ne disposez pas de suffisamment de données sur les interactions entre articles, vous pouvez utiliser Amazon Personalize pour collecter des données supplémentaires sur les événements en temps réel. À l'aide de certaines recettes et de certains cas d'utilisation, Amazon Personalize peut tirer des leçons de l'activité la plus récente de vos utilisateurs et mettre à jour les recommandations lorsqu'ils utilisent votre application.

Pour plus d'informations sur l'enregistrement des événements, notamment l'impact des événements sur les recommandations, une liste de services tiers de suivi des événements et des exemples de mise en œuvre, consultezEnregistrement des événements en temps réel pour influencer les recommandations.

Vos données sont-elles optimisées pour Amazon Personalize ?

Nous vous recommandons de vérifier les informations suivantes dans vos données :

- Vérifiez les valeurs manquantes. Nous recommandons qu'au moins 70 % de vos enregistrements contiennent des données pour chaque attribut. Nous recommandons que les colonnes qui autorisent les valeurs nulles soient remplies à au moins 70 %.
- Corrigez les inexactitudes ou les problèmes liés à vos données, tels que des conventions de dénomination incohérentes, des catégories dupliquées pour un élément, des incohérences IDs entre les ensembles de données ou des doublons. IDs Ces problèmes peuvent avoir un impact négatif sur les recommandations ou entraîner des comportements inattendus. Par exemple, vous pourriez avoir les deux « N/A” and “Not Applicable” in your data, but filter out recommendations based on only “N/A ». Les éléments marqués « Non applicable » ne seraient pas supprimés par le filtre.
- Si un élément, un utilisateur ou une action peut comporter plusieurs catégories, comme un film comportant plusieurs genres, combinez les valeurs catégorielles dans un seul attribut et séparez chaque valeur par l'opérateur |. Par exemple, les données GENRES d'un film peuvent être Action | Aventure | Thriller.
- Évitez d'avoir plus de 1 000 catégories possibles pour une colonne (sauf si la colonne contient des données uniquement à des fins de filtrage).

Pour obtenir la liste complète des recommandations relatives aux données et des instructions sur la façon dont vous pouvez utiliser Amazon Personalize pour identifier les problèmes, consultezAnalyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

Collectez-vous des données facultatives susceptibles d'améliorer les recommandations ?

Les données suivantes peuvent vous aider à améliorer la pertinence de vos recommandations.

- Type d'événement (obligatoire pour tous les cas d'utilisation de groupes de jeux de données de domaine)
- Valeur de l'événement
- Métadonnées contextuelles
- Métadonnées relatives aux articles et aux utilisateurs
- Données d'interaction entre les actions (utilisées uniquement par les recettes PERSONALIZED_ACTIONS)

Pour plus d'informations sur les types de données qu'Amazon Personalize peut utiliser, consultezTypes de données qu'Amazon Personalize peut utiliser.

Avez-vous un plan pour tester vos recommandations ?

Vous pouvez utiliser A/B les tests pour comparer les résultats de différents groupes d'utilisateurs interagissant avec les recommandations de différents modèles. A/B les tests peuvent vous aider à comparer différentes stratégies de recommandation et à voir si les recommandations vous aident à atteindre vos objectifs commerciaux. Pour de plus amples informations, veuillez consulter Mesurer l'impact des recommandations à l'aide A/B de tests.

Avez-vous d'autres objectifs commerciaux ?

Dans certains cas, vous pouvez avoir des objectifs en plus de générer des recommandations pertinentes pour vos utilisateurs. Par exemple, vous souhaiterez peut-être maximiser les revenus ou promouvoir certains types d'articles d'une certaine catégorie. Les fonctionnalités Amazon Personalize suivantes peuvent vous aider :

- Promotions : vous pouvez utiliser les promotions pour vous assurer qu'un certain pourcentage d'articles répond aux besoins de votre entreprise. Pour de plus amples informations, veuillez consulter Promouvoir des articles dans le cadre de recommandations en temps réel.
- Optimisation en fonction des objectifs commerciaux : pour certaines recettes de groupes de données personnalisés, vous pouvez optimiser une solution en fonction d'un objectif personnalisé, tel que la maximisation des minutes de diffusion ou l'augmentation des revenus. Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.
- Recommandations de filtrage. Utilisez des filtres pour appliquer des règles métier aux recommandations. Vous pouvez utiliser des filtres pour inclure ou exclure certains types d'éléments des recommandations. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Création d'un groupe de jeux de données Amazon Personalize

Après avoir créé des fichiers de schéma JSON pour vos données, vous êtes prêt à créer un groupe de jeux de données. Dans Amazon Personalize, un groupe de jeux de données est un conteneur pour les ressources Amazon Personalize, notamment les ensembles de données, les recommandations de domaines et les ressources personnalisées. Un groupe de jeux de données organise vos ressources en collections indépendantes, dans lesquelles les ressources d'un groupe de jeux de données ne peuvent influencer les ressources d'aucun autre groupe de jeux de données.

Vous créez un groupe de jeux de données pour chacun de vos domaines commerciaux. Par exemple, vous pouvez avoir une application qui fournit des recommandations pour le streaming vidéo et une autre qui fournit des recommandations pour les livres audio. Dans Amazon Personalize, vous devez créer un groupe de jeux de données pour chaque application. Ainsi, les données d'une application n'influencent pas les recommandations qu'Amazon Personalize génère pour l'autre application.

Vous pouvez créer un groupe de jeux de données de domaine ou un groupe de jeux de données personnalisé :

- Avec un groupe de jeux de données de domaine, vous créez des ressources préconfigurées et optimisées pour différents cas d'utilisation. Lorsque vous créez un groupe de jeux de données, vous en faites un groupe de données de domaine en spécifiant un domaine VIDEO_ON_DEMAND ou ECOMMERCE.

  Si vous possédez une application de streaming vidéo ou de commerce électronique, nous vous recommandons de créer un groupe de jeux de données de domaine. Vous pouvez toujours ajouter des ressources personnalisées, telles que des solutions et des versions de solutions conçues pour des cas d'utilisation personnalisés. Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine.
- Un groupe de jeux de données personnalisé inclut uniquement des ressources personnalisées que vous configurez en fonction de votre cas d'utilisation. Grâce à des ressources personnalisées, vous formez et déployez des solutions et des versions de solutions configurables (un modèle de recommandation Amazon Personalize formé) en fonction des besoins de votre entreprise. Si vous ne possédez pas d'application VIDEO_ON_DEMAND ou de commerce électronique, nous vous recommandons de créer un groupe de jeux de données personnalisé. Dans le cas contraire, nous

vous recommandons de commencer par un groupe de jeux de données de domaine et d'ajouter des ressources personnalisées si nécessaire.

Vous pouvez créer un groupe de jeux de données à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Rubriques

- Création d'un groupe de jeux de données (console)
- Création d'un groupe de jeux de données (AWS CLI)
- Création d'un groupe de jeux de données (AWS SDKs)

Création d'un groupe de jeux de données (console)

Créez un groupe de jeux de données en spécifiant le nom du groupe de données dans la console Amazon Personalize.

Pour créer un groupe de jeux de données

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez Create dataset group (Créer un groupe d'ensembles de données).
3. Si c'est la première fois que vous utilisez Amazon Personalize, sur la page Créer un groupe de jeux de données, dans Nouveau groupe de jeux de données, choisissez Get started.
4. Sous Dataset group details (Informations du groupe d'ensembles de données), pour Dataset group name (Nom du groupe d'ensembles de données), spécifiez un nom pour votre groupe d'ensembles de données.
5. Choisissez votre domaine :

   - Choisissez E-commerce pour créer un groupe de jeux de données de domaines ECOMMERCE.
   - Choisissez Vidéo à la demande pour créer un groupe de jeux de données de domaine VIDEO_ON_DEMAND.
   - Choisissez Personnalisé pour créer un groupe de jeux de données personnalisé contenant uniquement des ressources personnalisées, telles que des solutions, des campagnes et des tâches d'inférence par lots.

6. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
7. Choisissez Create dataset group (Créer un groupe d'ensembles de données). La page Vue d'ensemble s'affiche. Vous êtes maintenant prêt à créer un schéma et un jeu de données. Consultez Création d'un schéma et d'un jeu de données.

Création d'un groupe de jeux de données (AWS CLI)

Pour créer un groupe de jeux de données, utilisez l'`create-dataset-group`opération. Pour créer un groupe de jeux de données de domaine, pour le domaine, spécifiez `ECOMMERCE` ou`VIDEO_ON_DEMAND`. Pour créer un groupe de jeux de données personnalisé, ne spécifiez pas de domaine. Vous pouvez utiliser le paramètre Tags pour étiqueter éventuellement les ressources dans Amazon Personalize. Pour un exemple, voirAjouter des tags (AWS CLI).

Le code suivant crée un groupe de jeux de données de domaine pour le `VIDEO_ON_DEMAND` domaine.

```
aws personalize create-dataset-group \
--name dataset-group-name \
--domain VIDEO_ON_DEMAND
```

En cas de succès, le groupe de données Amazon Resource Name (ARN) du groupe de données s'affiche comme suit.

```
{
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
DatasetGroupName"
}
```

Enregistrez cette valeur pour une utilisation future. Pour afficher le groupe de jeux de données que vous avez créé, utilisez la `describe-dataset-group` commande et spécifiez l'ARN du groupe de jeux de données renvoyé.

```
aws personalize describe-dataset-group \
--dataset-group-arn dataset group arn
```

Le groupe de jeux de données et ses propriétés s'affichent comme suit.

```
{
    "datasetGroup": {
        "name": "DatasetGroupName",
        "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
DatasetGroupName",
        "status": "ACTIVE",
        "creationDateTime": 1542392161.262,
        "lastUpdatedDateTime": 1542396513.377
    }
}
```

Lorsque le groupe de jeux de données `status` est ACTIF, vous êtes prêt à créer un schéma et un jeu de données. Consultez Création d'un schéma et d'un jeu de données.

Création d'un groupe de jeux de données (AWS SDKs)

Utilisez le code suivant pour créer un groupe de jeux de données de domaine. Donnez un nom au groupe de jeux de données de domaine, et pour`domain`, spécifiez l'un `ECOMMERCE` ou l'autre`VIDEO_ON_DEMAND`. Pour créer un groupe de jeux de données personnalisé, modifiez le code pour supprimer le paramètre de domaine.

Pour plus d'informations sur le fonctionnement de l'API, consultez CreateDatasetGroup la section de référence de l'API. Vous pouvez utiliser le paramètre Tags pour étiqueter éventuellement les ressources dans Amazon Personalize. Pour un exemple, voirAjouter des tags (AWS SDKs).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
  name = 'dataset group name',
  domain = 'business domain'
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
```

```
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                               String datasetGroupName,
                                               String domain) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
 CreateDatasetGroupRequest.builder()
                .name(datasetGroupName)
                .domain(domain)
                .build();
        return
 personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: "NAME" /* required */,
  domain:
    "DOMAIN" /* required for a domain dsg, specify ECOMMERCE or VIDEO_ON_DEMAND */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetGroupCommand(domainDatasetGroupParams),
```

```
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

L'DescribeDatasetGroupopération renvoie le `datasetGroupArn` et le statut de l'opération. Lorsque le groupe de jeux de données `status` est ACTIF, vous êtes prêt à créer un schéma et un jeu de données. Consultez Création d'un schéma et d'un jeu de données.

Création d'un schéma et d'un jeu de données

Après avoir créé un groupe de jeux de données, vous êtes prêt à créer un schéma Amazon Personalize et un ensemble de données pour chaque type de données que vous importez. Un schéma indique à Amazon Personalize la structure de vos données et permet à Amazon Personalize d'analyser les données. Lorsque vous créez un schéma dans Amazon Personalize, vous utilisez le fichier JSON dans lequel vous l'avez crééCréation de fichiers JSON pour les schémas Amazon Personalize.

Un ensemble de données est un conteneur pour les données de formation dans Amazon Personalize. Les différents types de jeux de données ont des exigences différentes. Vous créez un ensemble de données pour chaque type de données que vous importez. Pour plus d'informations sur les différents types de jeux de données et sur la façon de préparer vos données, consultezPréparation des données de formation pour Amazon Personalize.

Vous pouvez créer des schémas et des ensembles de données avec la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou. AWS SDKs Vous ne pouvez pas créer les meilleures ressources d'action suivantes, notamment les ensembles de données d'actions et d'interactions d'action, dans un groupe de jeux de données de domaine.

> **Important**
>
> Une fois que vous avez créé un schéma, vous ne pouvez pas le modifier. Toutefois, si vous ajoutez de nouvelles colonnes, vous pouvez remplacer le schéma d'un jeu de données par un nouveau. Pour de plus amples informations, veuillez consulter Remplacement du schéma d'un ensemble de données pour ajouter de nouvelles colonnes.

Rubriques

- Création d'un jeu de données et d'un schéma (console)
- Création d'un jeu de données et d'un schéma (AWS CLI)
- Création d'un jeu de données et d'un schéma (AWS SDKs)

Création d'un jeu de données et d'un schéma (console)

S'il s'agit du premier jeu de données de votre groupe de jeux de données, votre premier type de jeu de données sera un jeu de données d'interactions entre éléments. Pour créer votre jeu de données d'interactions Item dans la console, spécifiez le nom du jeu de données, puis spécifiez un schéma JSON au format Avro. S'il ne s'agit pas du premier jeu de données de ce groupe de jeux de données, choisissez le type de jeu de données, puis spécifiez un nom et un schéma.

Pour plus d'informations sur les exigences relatives aux ensembles de données Amazon Personalize, consultezPréparation des données de formation pour Amazon Personalize. Si vous venez de terminer Création d'un groupe de jeux de données Amazon Personalize et que vous êtes déjà en train de créer votre jeu de données, passez à l'étape 4 de cette procédure.

Pour créer un jeu de données et un schéma

1. Si ce n'est pas déjà fait, suivez les instructions Création de fichiers JSON pour les schémas Amazon Personalize pour créer un fichier de schéma JSON qui décrit vos données.
2. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
3. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données dans lequel vous avez crééCréation d'un groupe de jeux de données Amazon Personalize.
4. À l'étape 1. Créez des ensembles de données et importez des données, choisissez Créer un ensemble de données et choisissez le type de jeu de données à créer.
5. Choisissez Importer les données directement dans les ensembles de données Amazon Personalize, puis Next.
6. Dans Détails du jeu de données, dans Nom du jeu de données, spécifiez le nom de votre ensemble de données.
7. Pour le schéma du jeu de données, choisissez Créer un nouveau schéma ou Utiliser un schéma existant.
8. Si vous utilisez un schéma existant, choisissez-le. Si vous créez un nouveau schéma, nommez-le et collez-y le schéma JSON correspondant à vos données. Vous avez créé ce fichier dansCréation de fichiers JSON pour les schémas Amazon Personalize.
9. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
10. Choisissez Next et suivez les instructions Importation de données de formation dans les ensembles de données Amazon Personalize pour importer vos données.

Création d'un jeu de données et d'un schéma (AWS CLI)

Pour créer un ensemble de données et un schéma à l'aide de AWS CLI, vous devez utiliser la `create-schema` commande (qui utilise l'opération d'CreateSchemaAPI) puis `create-dataset` (qui utilise l'opération d'CreateDatasetAPI).

Pour créer un schéma et un jeu de données

1. Si ce n'est pas déjà fait, suivez les instructions Création de fichiers JSON pour les schémas Amazon Personalize pour créer un fichier de schéma JSON qui décrit vos données.
2. Créez un schéma dans Amazon Personalize en exécutant la commande suivante. Une fois que vous avez créé un schéma, vous ne pouvez pas le modifier. Remplacez `schemaName` par le nom du schéma, puis `file://SchemaName.json` par l'emplacement de votre fichier JSON. L'exemple montre que le fichier appartient au dossier actif. Si vous créez une structure pour un jeu de données dans un groupe de jeux de données de domaine, ajoutez le `domain` paramètre et définissez-le sur `ECOMMERCE` ou`VIDEO_ON_DEMAND`. Pour plus d'informations sur l'API, consultez CreateSchema.

   ```
   aws personalize create-schema \
     --name SchemaName \
     --schema file://SchemaName.json
   ```

   Le schéma Amazon Resource Name (ARN) s'affiche, comme illustré dans l'exemple suivant :

   ```
   {
     "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/SchemaName"
   }
   ```

3. Créez un ensemble de données vide en exécutant la commande suivante. Indiquez le nom de ressource Amazon (ARN) du groupe de données Création d'un groupe de jeux de données (AWS CLI) et l'ARN du schéma de l'étape précédente. Les valeurs de type d'ensemble de données peuvent être `Interactions` `UsersItems`,`Actions`,, ou`Action_Interactions`. Pour plus d'informations sur l'API, consultez CreateDataset.

   ```
   aws personalize create-dataset \
     --name Dataset Name \
     --dataset-group-arn Dataset Group ARN \
     --dataset-type Dataset Type \
   ```

```
  --schema-arn Schema Arn
```

L'ARN du jeu de données est affiché, comme indiqué dans l'exemple suivant.

```
{
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/DatasetName/
INTERACTIONS"
}
```

4. Enregistrez l'ARN du jeu de données pour une utilisation ultérieure. Après avoir créé un jeu de données, vous êtes prêt à importer vos données d'entraînement. Consultez Importation de données de formation dans les ensembles de données Amazon Personalize.

Création d'un jeu de données et d'un schéma (AWS SDKs)

Pour créer un ensemble de données et un schéma à l'aide de AWS SDKs, vous devez d'abord définir un schéma au format Avro et l'ajouter à Amazon Personalize à l'aide de l'CreateSchemaopération. Une fois que vous avez créé un schéma, vous ne pouvez pas le modifier. Créez ensuite un ensemble de données à l'aide de l'CreateDatasetopération.

Pour créer un schéma et un jeu de données

1. Si ce n'est pas déjà fait, suivez les instructions Création de fichiers JSON pour les schémas Amazon Personalize pour créer un fichier de schéma JSON qui décrit vos données.
2. Créez un schéma dans Amazon Personalize avec le code suivant. Spécifiez le nom de votre schéma et le chemin du fichier JSON de schéma. Si vous créez une structure pour un jeu de données dans un groupe de jeux de données de domaine, ajoutez le `domain` paramètre et définissez-le sur `ECOMMERCE` ou`VIDEO_ON_DEMAND`. Pour plus d'informations sur l'API, consultez CreateSchema.

   SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

with open('schemaFile.json') as f:
    createSchemaResponse = personalize.create_schema(
        name = 'schema name',
```

```
        schema = f.read()
    )

schema_arn = createSchemaResponse['schemaArn']

print('Schema ARN:' + schema_arn )
```

SDK for Java 2.x

```
public static String createSchema(PersonalizeClient personalizeClient, String
 schemaName, String filePath) {

    String schema = null;

    try {
        schema = new String(Files.readAllBytes(Paths.get(filePath)));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    try {
        CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
                .name(schemaName)
                .schema(schema)
                .build();

        String schemaArn =
 personalizeClient.createSchema(createSchemaRequest).schemaArn();
        System.out.println("Schema arn: " + schemaArn);

        return schemaArn;

    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from "@aws-sdk/client-personalize";
```

```
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from "node:fs";

const schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = "TEST"; // For unit tests.
}
// Set the schema parameters.
export const createSchemaParam = {
  name: "NAME" /* required */,
  schema: mySchema /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSchemaCommand(createSchemaParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Amazon Personalize renvoie l'ARN du nouveau schéma. Enregistrez-le car vous en aurez besoin à l'étape suivante.

3. Créez un ensemble de données à l'aide de l'CreateDatasetopération. Le code suivant montre comment créer un ensemble de données. Spécifiez le nom de ressource Amazon (ARN) de votre groupe de jeux de données, l'ARN du schéma de l'étape précédente et spécifiez le type de jeu de données. Les valeurs de type d'ensemble de données peuvent être `Interactions` `UsersItems`,`Actions`,, ou`Action_Interactions`. Pour plus d'informations sur les différents

types de jeux de données, consultezPréparation des données de formation pour Amazon Personalize.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset(
    name = 'dataset_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'dataset_type'
)

print ('Dataset Arn: ' + response['datasetArn'])
```

SDK for Java 2.x

```
public static String createDataset(PersonalizeClient personalizeClient,
                                   String datasetName,
                                   String datasetGroupArn,
                                   String datasetType,
                                   String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
                .name(datasetName)
                .datasetGroupArn(datasetGroupArn)
                .datasetType(datasetType)
                .schemaArn(schemaArn).build();

        String datasetArn =
 personalizeClient.createDataset(request).datasetArn();
        System.out.println("Dataset " + datasetName + " created. Dataset ARN: "
 + datasetArn);

        return datasetArn;

    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
```

```
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  datasetType: "DATASET_TYPE" /* required */,
  name: "NAME" /* required */,
  schemaArn: "SCHEMA_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetCommand(createDatasetParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Après avoir créé un jeu de données, vous êtes prêt à importer vos données d'entraînement. Consultez Importation de données de formation dans les ensembles de données Amazon Personalize.

Importation de données de formation dans les ensembles de données Amazon Personalize

Une fois que vous avez créé un schéma et un jeu de données, vous êtes prêt à importer vos données d'entraînement dans le jeu de données. Lorsque vous importez des données, vous pouvez choisir d'importer des enregistrements en bloc, individuellement ou les deux.

- Les importations en vrac impliquent l'importation d'un grand nombre de documents historiques à la fois. Vous pouvez préparer vous-même des données en masse et les importer directement dans Amazon Personalize à partir d'un fichier CSV dans Amazon S3. Pour plus d'informations sur la préparation de vos données, consultezPréparation des données de formation pour Amazon Personalize. Si vous avez besoin d'aide pour préparer vos données, vous pouvez utiliser SageMaker AI Data Wrangler pour préparer et importer vos interactions avec les articles en vrac, ainsi que les données relatives aux utilisateurs et aux articles. Pour de plus amples informations, veuillez consulter Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler.
- Si vous ne disposez pas de données en masse, vous pouvez utiliser des opérations d'importation individuelles pour collecter des données et diffuser des événements jusqu'à ce que vous répondiez aux exigences de formation d'Amazon Personalize et aux exigences en matière de données de votre cas d'utilisation ou de votre recette de domaine. Pour plus d'informations sur l'enregistrement d'événements, consultezEnregistrement des événements en temps réel pour influencer les recommandations. Pour plus d'informations sur l'importation d'enregistrements individuels, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize.

Après avoir importé des données dans un ensemble de données Amazon Personalize, vous pouvez les analyser, les exporter vers un compartiment Amazon S3, les mettre à jour ou les supprimer en supprimant le jeu de données.

Si vous importez un article, un utilisateur ou une action avec le même identifiant qu'un enregistrement figurant déjà dans votre ensemble de données, Amazon Personalize le remplace par le nouvel enregistrement. Si vous enregistrez deux événements d'interaction entre un article ou une action avec exactement le même horodatage et les mêmes propriétés, Amazon Personalize ne conserve qu'un seul des événements.

Au fur et à mesure que votre catalogue s'agrandit, mettez à jour vos données historiques avec des opérations d'importation de données groupées ou individuelles supplémentaires. Pour obtenir des recommandations en temps réel, maintenez votre jeu de données sur les interactions entre les articles à jour en fonction du comportement de vos utilisateurs. Pour ce faire, vous enregistrez les événements d'interaction en temps réel avec un outil de suivi des événements et l'PutEventsopération. Pour de plus amples informations, consultez Enregistrement des événements en temps réel pour influencer les recommandations.

Après avoir importé vos données, vous êtes prêt à créer des recommandations de domaines (pour les groupes de jeux de données de domaine) ou des ressources personnalisées (pour les groupes de données personnalisés) afin d'entraîner un modèle sur vos données. Vous utilisez ces ressources pour générer des recommandations. Pour plus d'informations, consultez Recommandeurs de domaines dans Amazon Personalize ou Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Rubriques

- Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données
- Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler
- Importation d'enregistrements individuels dans un ensemble de données Amazon Personalize

Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données

Une fois que vous avez formaté vos données d'entrée (voirPréparation des données de formation pour Amazon Personalize) et que vous avez terminéCréation d'un schéma et d'un jeu de données, vous êtes prêt à importer vos données en masse à l'aide d'une tâche d'importation de jeux de données. Une tâche d'importation de jeu de données est un outil d'importation en masse qui remplit un ensemble de données provenant d'Amazon S3.

Pour importer des données depuis Amazon S3, vos fichiers CSV doivent se trouver dans un compartiment Amazon S3 et vous devez autoriser Amazon Personalize à accéder à vos ressources Amazon S3 :

- Pour plus d'informations sur le chargement de fichiers vers Amazon S3, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service.

- Pour plus d'informations sur la façon d'autoriser Amazon Personalize à accéder à vos fichiers dans Amazon S3, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.

  Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Vous pouvez créer une tâche d'importation de jeux de données à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Si vous avez déjà créé une tâche d'importation de jeu de données pour un ensemble de données, vous pouvez utiliser une nouvelle tâche d'importation de jeu de données pour ajouter ou remplacer les données en bloc existantes. Pour de plus amples informations, veuillez consulter Mise à jour des données dans les ensembles de données après l'entraînement.

Si vous importez un article, un utilisateur ou une action avec le même identifiant qu'un enregistrement figurant déjà dans votre ensemble de données, Amazon Personalize le remplace par le nouvel enregistrement. Si vous enregistrez deux événements d'interaction entre un article ou une action avec exactement le même horodatage et les mêmes propriétés, Amazon Personalize ne conserve qu'un seul des événements.

Après avoir importé vos données, vous êtes prêt à créer des recommandations de domaines (pour les groupes de jeux de données de domaine) ou des ressources personnalisées (pour les groupes de données personnalisés) afin d'entraîner un modèle sur vos données. Vous utilisez ces ressources pour générer des recommandations. Pour plus d’informations, consultez Recommandeurs de domaines dans Amazon Personalize ou Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Rubriques

- Modes d'importation
- Création d'une tâche d'importation de jeux de données (console)
- Création d'une tâche d'importation de jeux de données (AWS CLI)
- Création d'une tâche d'importation de jeux de données (AWS SDKs)

Modes d'importation

Si vous avez déjà créé une tâche d'importation pour l'ensemble de données, vous pouvez configurer la manière dont Amazon Personalize ajoute vos nouveaux enregistrements. Pour ce faire, vous devez spécifier un mode d'importation pour votre tâche d'importation de jeux de données. Si vous n'avez pas importé d'enregistrements groupés, le champ Mode d'importation n'est pas disponible dans la console et vous ne pouvez le spécifier que `FULL` dans le cadre du fonctionnement de l'`CreateDatasetImportJob`API. Par défaut, il s'agit d'un remplacement complet.

- Pour remplacer toutes les données en masse existantes dans votre ensemble de données, choisissez Remplacer les données existantes dans la console Amazon Personalize ou spécifiez-les `FULL` dans l'opération CreateDatasetImportJob d'API. Cela ne remplace pas les données que vous avez importées individuellement, y compris les événements enregistrés en temps réel.
- Pour ajouter les enregistrements aux données existantes de votre ensemble de données, choisissez Ajouter aux données existantes ou spécifiez-les `INCREMENTAL` dans l'opération `CreateDatasetImportJob` d'API. Amazon Personalize remplace tout enregistrement portant le même identifiant par le nouveau.

> **Note**
>
> Pour ajouter des données à un jeu de données d'interactions d'éléments ou à un jeu de données d'interactions d'action avec une tâche d'importation de jeu de données, vous devez disposer d'au moins 1 000 nouveaux enregistrements d'interactions entre éléments ou d'interactions d'action.

Création d'une tâche d'importation de jeux de données (console)

> **Important**
>
> Par défaut, une tâche d'importation de jeu de données remplace toutes les données existantes du jeu de données que vous avez importé en bloc. Si vous avez déjà importé des données en masse, vous pouvez ajouter des données en modifiant le mode d'importation de la tâche.

Pour importer des enregistrements en masse dans un ensemble de données à l'aide de la console Amazon Personalize, créez une tâche d'importation d'ensemble de données avec un nom, le rôle de service IAM et l'emplacement de vos données.

Si vous venez de créer votre jeu de données dansCréation d'un schéma et d'un jeu de données, passez à l'étape 5.

Pour importer des enregistrements en bloc (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données. L'aperçu du groupe de jeux de données s'affiche.
3. Dans le volet de navigation, choisissez Datasets et choisissez le jeu de données dans lequel vous souhaitez importer des données en masse.
4. Dans Tâches d'importation de jeux de données, choisissez Créer une tâche d'importation de jeu de données.
5. S'il s'agit de votre première tâche d'importation de jeux de données, dans Source d'importation de données, sélectionnez Importer des données depuis S3.
6. Pour le nom de la tâche d'importation du jeu de données, spécifiez un nom pour votre tâche d'importation.
7. Si vous avez déjà importé des données en masse, dans Mode d'importation, choisissez le mode de mise à jour du jeu de données. Choisissez Remplacer les données existantes ou Ajouter aux données existantes. données. Cette option n'apparaît pas s'il s'agit de votre première tâche pour l'ensemble de données. Pour de plus amples informations, veuillez consulter Mise à jour des données dans les ensembles de données après l'entraînement.
8. Dans Source d'importation des données, pour Emplacement des données, spécifiez où votre fichier de données est stocké dans Amazon S3. Utilisez la syntaxe suivante :

   ```
   s3:/amzn-s3-demo-bucket/<folder path>/<CSV filename>
   ```

   Si vos fichiers CSV se trouvent dans un dossier de votre compartiment Amazon S3 et que vous souhaitez charger plusieurs fichiers CSV dans un ensemble de données avec une seule tâche d'importation d'ensemble de données, vous pouvez spécifier le chemin d'accès au dossier. Amazon Personalize utilise uniquement les fichiers du premier niveau de votre dossier, aucune donnée n'est utilisée dans aucun sous-dossier. Utilisez la syntaxe suivante avec un / après le nom du dossier :

```
s3:/amzn-s3-demo-bucket/<folder path>/
```

9. Dans le rôle IAM, choisissez de créer un nouveau rôle ou d'utiliser un rôle existant. Si vous avez rempli les conditions requises, choisissez Utiliser un rôle de service existant et spécifiez le rôle que vous avez créé dansCréation d'un rôle IAM pour Amazon Personalize.
10. Si vous avez créé une attribution de métrique et que vous souhaitez publier des métriques liées à cette tâche sur Amazon S3, dans Publier les métriques d'événements sur S3, choisissez Publier les métriques pour cette tâche d'importation.

    Si vous n'en avez pas créé une et que vous souhaitez publier des statistiques pour cette tâche, choisissez Create metric attribution pour en créer une nouvelle dans un autre onglet. Après avoir créé l'attribution des métriques, vous pouvez revenir à cet écran et terminer la création de la tâche d'importation.

    Pour plus d'informations sur les attributions métriques, voirMesurer l'impact des recommandations d'Amazon Personalize.
11. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
12. Choisissez Start import (Démarrer l'importation). La tâche d'importation des données démarre et la page Dashboard Overview (Présentation du tableau de bord) s'affiche. L'importation d'ensembles de données est terminée quand le statut ACTIVE (ACTIF) apparaît. Après avoir importé des données dans un ensemble de données Amazon Personalize, vous pouvez les analyser, les exporter vers un compartiment Amazon S3, les mettre à jour ou les supprimer en supprimant le jeu de données.

    Après avoir importé vos données, vous êtes prêt à créer des recommandations de domaines (pour les groupes de jeux de données de domaine) ou des ressources personnalisées (pour les groupes de données personnalisés) afin d'entraîner un modèle sur vos données. Vous utilisez ces ressources pour générer des recommandations. Pour plus d'informations, consultez Recommandeurs de domaines dans Amazon Personalize ou Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Création d'une tâche d'importation de jeux de données (AWS CLI)

> **Important**
>
> Par défaut, une tâche d'importation de jeu de données remplace toutes les données existantes du jeu de données que vous avez importé en bloc. Si vous avez déjà importé des données en masse, vous pouvez ajouter des données en modifiant le mode d'importation de la tâche.

Pour importer des enregistrements en bloc à l'aide du AWS CLI, créez une tâche d'importation de jeu de données à l'aide de la CreateDatasetImportJob commande. Si vous avez déjà créé une tâche d'importation d'ensemble de données pour un ensemble de données, vous pouvez utiliser le paramètre du mode d'importation pour spécifier comment ajouter les nouvelles données. Pour plus d'informations sur la mise à jour des données en bloc existantes, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Importer des enregistrements en bloc (AWS CLI)

1. Créez une tâche d'importation d'ensembles de données en exécutant la commande suivante. Indiquez le nom de ressource Amazon (ARN) de votre ensemble de données et spécifiez le chemin d'accès au compartiment Amazon S3 dans lequel vous avez stocké les données d'entraînement. Utilisez la syntaxe suivante pour le chemin :

   ```
   s3:/amzn-s3-demo-bucket/<folder path>/<CSV filename>
   ```

   Si vos fichiers CSV se trouvent dans un dossier de votre compartiment Amazon S3 et que vous souhaitez charger plusieurs fichiers CSV dans un ensemble de données avec une seule tâche d'importation d'ensemble de données, vous pouvez spécifier le chemin d'accès au dossier. Amazon Personalize utilise uniquement les fichiers du premier niveau de votre dossier, aucune donnée n'est utilisée dans aucun sous-dossier. Utilisez la syntaxe suivante avec un `/` après le nom du dossier :

   ```
   s3:/amzn-s3-demo-bucket/<folder path>/
   ```

   Indiquez le rôle Gestion des identités et des accès AWS (IAM) Amazon Resource Name (ARN) dans Création d'un rôle IAM pour Amazon Personalize lequel vous avez créé. L'`import-mode` par défaut est `FULL`. Pour de plus amples informations, veuillez consulter Mise à jour des

données dans les ensembles de données après l'entraînement. Pour plus d'informations sur ce fonctionnement, consultez CreateDatasetImportJob.

```
aws personalize create-dataset-import-job \
--job-name dataset import job name \
--dataset-arn dataset arn \
--data-source dataLocation=s3://amzn-s3-demo-bucket/filename \
--role-arn roleArn \
--import-mode FULL
```

L'ARN de la tâche d'importation de l'ensemble de données est affiché, comme indiqué dans l'exemple suivant.

```
{
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/
DatasetImportJobName"
}
```

2. Vérifiez le statut à l'aide de la commande `describe-dataset-import-job`. Fournissez l'ARN de la tâche d'importation d'ensembles de données qui a été retourné lors de l'étape précédente. Pour plus d'informations sur ce fonctionnement, consultez DescribeDatasetImportJob.

```
aws personalize describe-dataset-import-job \
--dataset-import-job-arn dataset import job arn
```

Les propriétés de la tâche d'importation d'ensembles de données et son statut sont affichés. Au départ, il s'`status`affiche sous la forme CREATE PENDING.

```
{
  "datasetImportJob": {
      "jobName": "Dataset Import job name",
      "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-
job/DatasetImportJobArn",
      "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
DatasetGroupName/INTERACTIONS",
      "dataSource": {
          "dataLocation": "s3://amzn-s3-demo-bucket/ratings.csv"
      },
      "importMode": "FULL",
      "roleArn": "role-arn",
      "status": "CREATE PENDING",
```

```
        "creationDateTime": 1542392161.837,
        "lastUpdatedDateTime": 1542393013.377
    }
}
```

L'importation d'ensembles de données est terminée quand le statut ACTIVE (ACTIF) apparaît. Après avoir importé des données dans un ensemble de données Amazon Personalize, vous pouvez les analyser, les exporter vers un compartiment Amazon S3, les mettre à jour ou les supprimer en supprimant le jeu de données.

Après avoir importé vos données, vous êtes prêt à créer des recommandations de domaines (pour les groupes de jeux de données de domaine) ou des ressources personnalisées (pour les groupes de données personnalisés) afin d'entraîner un modèle sur vos données. Vous utilisez ces ressources pour générer des recommandations. Pour plus d'informations, consultez Recommandeurs de domaines dans Amazon Personalize ou Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Création d'une tâche d'importation de jeux de données (AWS SDKs)

> **Important**
>
> Par défaut, une tâche d'importation de jeu de données remplace toutes les données existantes du jeu de données que vous avez importé en bloc. Si vous avez déjà importé des données en masse, vous pouvez ajouter des données en modifiant le mode d'importation de la tâche.

Pour importer des données, créez une tâche d'importation de jeu de données avec l'CreateDatasetImportJobopération. Le code suivant montre comment créer une tâche d'importation de jeu de données.

Donnez le nom de la tâche, définissez `datasetArn` le nom de ressource Amazon (ARN) de votre ensemble de données et définissez le `dataLocation` chemin d'accès au compartiment Amazon S3 dans lequel vous avez stocké les données d'entraînement. Utilisez la syntaxe suivante pour le chemin :

`s3:/amzn-s3-demo-bucket/<folder path>/<CSV filename>.csv`

Si vos fichiers CSV se trouvent dans un dossier de votre compartiment Amazon S3 et que vous souhaitez charger plusieurs fichiers CSV dans un ensemble de données avec une seule tâche d'importation d'ensemble de données, vous pouvez spécifier le chemin d'accès au dossier. Amazon Personalize utilise uniquement les fichiers du premier niveau de votre dossier, aucune donnée n'est utilisée dans aucun sous-dossier. Utilisez la syntaxe suivante avec un `/` après le nom du dossier :

`s3:/amzn-s3-demo-bucket/<folder path>/`

Pour cela`roleArn`, spécifiez le rôle Gestion des identités et des accès AWS (IAM) qui autorise Amazon Personalize à accéder à votre compartiment S3. Consultez Création d'un rôle IAM pour Amazon Personalize. L'`importMode` par défaut est `FULL`. Cela remplace toutes les données en masse de l'ensemble de données. Pour ajouter des données, définissez-les sur. `INCREMENTAL` Pour plus d'informations sur la mise à jour des données en bloc existantes, consultezMise à jour des données dans les ensembles de données après l'entraînement.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation':'s3://amzn-s3-demo-bucket/filename.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
 personalizeClient,
                                                        String jobName,
                                                        String datasetArn,
                                                        String s3BucketPath,
                                                        String roleArn,
                                                        ImportMode importMode) {

  long waitInMilliseconds = 60 * 1000;
  String status;
  String datasetImportJobArn;

  try {
      DataSource importDataSource = DataSource.builder()
              .dataLocation(s3BucketPath)
              .build();

      CreateDatasetImportJobRequest createDatasetImportJobRequest =
 CreateDatasetImportJobRequest.builder()
              .datasetArn(datasetArn)
              .dataSource(importDataSource)
              .jobName(jobName)
              .roleArn(roleArn)
              .importMode(importMode)
              .build();

      datasetImportJobArn =
 personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
              .datasetImportJobArn();

      DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
 DescribeDatasetImportJobRequest.builder()
              .datasetImportJobArn(datasetImportJobArn)
              .build();

      long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

      while (Instant.now().getEpochSecond() < maxTime) {

          DatasetImportJob datasetImportJob = personalizeClient
                  .describeDatasetImportJob(describeDatasetImportJobRequest)
                  .datasetImportJob();
```

```
            status = datasetImportJob.status();
            System.out.println("Dataset import job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return datasetImportJobArn;

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN', /* required */
  dataSource: {
    dataLocation: 's3://amzn-s3-demo-bucket/<folderName>/<CSVfilename>.csv'  /*
 required */
  },
  jobName: 'NAME',           /* required */
  roleArn: 'ROLE_ARN',       /* required */
  importMode: "FULL"         /* optional, default is FULL */
};
```

```
export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

La réponse de l'DescribeDatasetImportJobopération inclut le statut de l'opération.

Vous devez attendre que l'état passe à ACTIVE avant de pouvoir utiliser les données pour former un modèle.

L'importation d'ensembles de données est terminée quand le statut ACTIVE (ACTIF) apparaît. Après avoir importé des données dans un ensemble de données Amazon Personalize, vous pouvez les analyser, les exporter vers un compartiment Amazon S3, les mettre à jour ou les supprimer en supprimant le jeu de données.

Après avoir importé vos données, vous êtes prêt à créer des recommandations de domaines (pour les groupes de jeux de données de domaine) ou des ressources personnalisées (pour les groupes de données personnalisés) afin d'entraîner un modèle sur vos données. Vous utilisez ces ressources pour générer des recommandations. Pour plus d’informations, consultez Recommandeurs de domaines dans Amazon Personalize ou Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler

> **Important**
>
> Lorsque vous utilisez Data Wrangler, vous encourez des coûts liés à l'IA. SageMaker Pour obtenir la liste complète des frais et des prix, consultez l'onglet Data Wrangler des tarifs d'Amazon SageMaker AI. Pour éviter d'encourir des frais supplémentaires, arrêtez votre

instance Data Wrangler lorsque vous avez terminé. Pour plus d'informations, voir Arrêter Data Wrangler.

Après avoir créé un groupe de jeux de données, vous pouvez utiliser Amazon SageMaker AI Data Wrangler (Data Wrangler) pour importer des données provenant de plus de 40 sources dans un ensemble de données Amazon Personalize. Data Wrangler est une fonctionnalité d'Amazon SageMaker AI Studio Classic qui fournit une solution de bout en bout pour importer, préparer, transformer et analyser des données. Vous ne pouvez pas utiliser Data Wrangler pour préparer et importer des données dans un ensemble de données Actions ou Action interactions.

Lorsque vous utilisez Data Wrangler pour préparer et importer des données, vous utilisez un flux de données. Un flux de données définit une série d'étapes de préparation des données d'apprentissage automatique, en commençant par l'importation des données. Chaque fois que vous ajoutez une étape à votre flux, Data Wrangler agit sur vos données, par exemple en les transformant ou en générant une visualisation.

Voici certaines des étapes que vous pouvez ajouter à votre flux pour préparer les données pour Amazon Personalize :

- Informations : vous pouvez ajouter des étapes d'analyse spécifiques à Amazon Personalize à votre flux. Ces informations peuvent vous aider à mieux connaître vos données et les mesures que vous pouvez prendre pour les améliorer.
- Visualisations : vous pouvez ajouter des étapes de visualisation pour générer des graphiques tels que des histogrammes et des nuages de points. Les graphiques peuvent vous aider à détecter les problèmes liés à vos données, tels que les valeurs aberrantes ou les valeurs manquantes.
- Transformations : vous pouvez utiliser les étapes de transformation spécifiques et générales d'Amazon Personalize pour vous assurer que vos données répondent aux exigences d'Amazon Personalize. La transformation Amazon Personalize vous permet de mapper vos colonnes de données aux colonnes requises en fonction du type de jeu de données Amazon Personalize.

Si vous devez quitter Data Wrangler avant d'importer des données dans Amazon Personalize, vous pouvez revenir là où vous vous êtes arrêté en choisissant le même type de jeu de données lorsque vous lancez Data Wrangler depuis la console Amazon Personalize. Vous pouvez également accéder à Data Wrangler directement via SageMaker AI Studio Classic.

Nous vous recommandons d'importer les données de Data Wrangler dans Amazon Personalize comme suit. Les étapes de transformation, de visualisation et d'analyse sont facultatives, répétables et peuvent être effectuées dans n'importe quel ordre.

1. Configurer les autorisations : configurez les autorisations pour Amazon Personalize et les rôles de service SageMaker AI. Et configurez des autorisations pour vos utilisateurs.
2. Lancez Data Wrangler dans SageMaker AI Studio Classic depuis la console Amazon Personalize : utilisez la console Amazon Personalize pour configurer un domaine SageMaker AI et lancez Data Wrangler dans SageMaker AI Studio Classic.
3. Importez vos données dans Data Wrangler - Importez des données provenant de plus de 40 sources dans Data Wrangler. Les sources incluent AWS des services, tels qu'Amazon Redshift, Amazon EMR ou Amazon Athena, et des tiers tels que Snowflake ou. DataBricks
4. Transformez vos données : utilisez Data Wrangler pour transformer vos données afin de répondre aux exigences d'Amazon Personalize.
5. Visualisez et analysez vos données : utilisez Data Wrangler pour visualiser vos données et les analyser grâce à des informations spécifiques d'Amazon Personalize.
6. Traitez et importez des données dans Amazon Personalize : utilisez un bloc-notes Jupyter SageMaker AI Studio Classic pour importer vos données traitées dans Amazon Personalize.

Informations supplémentaires

Les ressources suivantes fournissent des informations supplémentaires sur l'utilisation d'Amazon SageMaker AI Data Wrangler et d'Amazon Personalize.

- Pour un didacticiel expliquant le traitement et la transformation d'un exemple de jeu de données, consultez Demo : Data Wrangler Titanic Dataset Walkthrough dans le guide du développeur Amazon SageMaker AI. Ce didacticiel présente les champs et les fonctions de Data Wrangler.
- Pour plus d'informations sur l'intégration aux domaines Amazon SageMaker AI, consultez la section Intégration rapide au domaine Amazon SageMaker AI dans le manuel Amazon SageMaker AI Developer Guide.
- Pour plus d'informations sur les exigences relatives aux données d'Amazon Personalize, consultezPréparation des données de formation pour Amazon Personalize.

Configurer les autorisations

Pour préparer les données avec Data Wrangler, vous devez configurer les autorisations suivantes :

- Créez un rôle de service pour Amazon Personalize : si ce n'est pas déjà fait, suivez les instructions Configuration d'Amazon Personalize pour créer un rôle de service IAM pour Amazon Personalize. Ce rôle doit comporter `GetObject` des `ListBucket` autorisations pour les compartiments Amazon S3 qui stockent vos données traitées. Et il doit être autorisé à utiliser n'importe quelle AWS KMS clé.

  Pour plus d'informations sur l'octroi à Amazon Personalize de l'accès à vos compartiments Amazon S3, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3. Pour plus d'informations sur l'octroi à Amazon Personalize de l'accès à vos AWS KMS clés, consultezAutoriser Amazon Personalize à utiliser votre AWS KMS clé.
- Créez un utilisateur administratif avec des autorisations SageMaker AI : votre administrateur doit avoir un accès complet à l' SageMaker IA et doit être en mesure de créer un domaine SageMaker AI. Pour plus d'informations, consultez la section Créer un utilisateur et un groupe administratifs dans le manuel Amazon SageMaker AI Developer Guide.
- Création d'un rôle d'exécution SageMaker AI : créez un rôle d'exécution SageMaker AI avec accès aux ressources d' SageMaker IA et aux opérations d'importation de données Amazon Personalize. La `AmazonSageMakerFullAccess`politique doit être attachée au rôle d'exécution de l' SageMaker IA. Si vous avez besoin d'autorisations Data Wrangler plus détaillées, consultez Data Wrangler Security and Permissions dans le manuel Amazon AI Developer Guide. SageMaker Pour plus d'informations sur les rôles d' SageMaker IA, consultez la section Rôles d'SageMaker IA.

  Pour accorder l'accès aux opérations d'importation de données Amazon Personalize, associez la politique IAM suivante au rôle d'exécution SageMaker AI. Cette politique accorde les autorisations requises pour importer des données dans Amazon Personalize et associer une politique à votre compartiment Amazon S3. Et il accorde des `PassRole` autorisations lorsque le service est Amazon Personalize. Mettez à jour Amazon S3 `amzn-s3-demo-bucket` avec le nom du compartiment Amazon S3 que vous souhaitez utiliser comme destination pour vos données formatées après les avoir préparées avec Data Wrangler.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "personalize:Create*",
                "personalize:List*",
                "personalize:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "personalize.amazonaws.com"
                }
            }
        }
    ]
}
```

Pour plus d'informations sur la création d'une stratégie IAM, consultez la section Création de politiques IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'attachement d'une politique IAM à un rôle, consultez la section Ajouter et supprimer des autorisations d'identité IAM dans le guide de l'utilisateur IAM.

Lancement de Data Wrangler depuis Amazon Personalize

Pour lancer Data Wrangler depuis Amazon Personalize, vous devez utiliser la console Amazon Personalize pour configurer un domaine SageMaker AI et lancer Data Wrangler.

Pour lancer Data Wrangler depuis Amazon Personalize

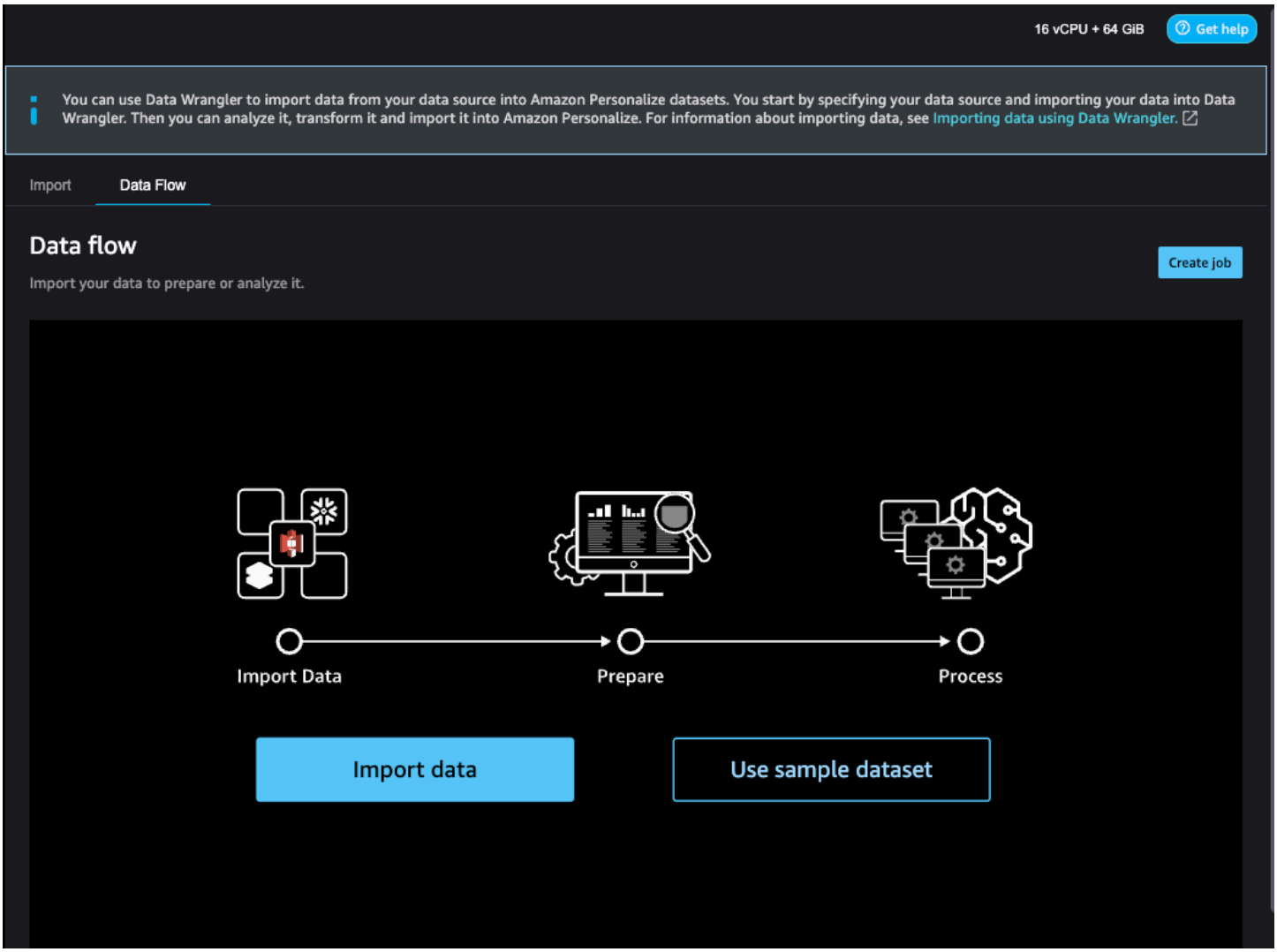
1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Dans Configurer des ensembles de données, choisissez Créer un ensemble de données et choisissez le type de jeu de données à créer. Vous ne pouvez pas utiliser Data Wrangler pour préparer un ensemble de données Actions ou Action interactions.
4. Choisissez Importer des données à l'aide de Data Wrangler, puis cliquez sur Suivant.
5. Pour le SageMaker domaine, choisissez d'utiliser un domaine existant ou d'en créer un nouveau. Vous avez besoin d'un domaine SageMaker AI pour accéder à Data Wrangler dans SageMaker AI Studio Classic. Pour plus d'informations sur les domaines et les profils utilisateur, consultez SageMaker AI Domain dans le manuel Amazon SageMaker AI Developer Guide.
6. Pour utiliser un domaine existant, choisissez un domaine SageMaker AI et un profil utilisateur pour configurer le domaine.
7. Pour créer un nouveau domaine :
   - Donnez un nom au nouveau domaine.
   - Choisissez un nom de profil utilisateur.
   - Pour Rôle d'exécution, choisissez le rôle dans lequel vous l'avez crééConfigurer les autorisations. Ou, si vous disposez d' CreateRole autorisations, créez un nouveau rôle à l'aide de l'assistant de création de rôle. La `AmazonSageMakerFullAccess` politique doit être attachée au rôle que vous utilisez.
8. Choisissez Suivant. Si vous créez un nouveau domaine, SageMaker AI commence à créer votre domaine. Cela peut prendre jusqu'à dix minutes.
9. Vérifiez les détails de votre domaine SageMaker AI.
10. Choisissez Importer des données avec Data Wrangler. SageMaker AI Studio Classic commence à créer votre environnement, et une fois terminée, la page de flux de données de Data Wrangler dans SageMaker AI Studio Classic s'ouvre dans un nouvel onglet. SageMaker AI Studio Classic peut mettre jusqu'à cinq minutes pour terminer la création de votre environnement. Une fois

l'opération terminée, vous êtes prêt à commencer à importer des données dans Data Wrangler. Pour plus d'informations, consultez Importation de données dans Data Wrangler.

Importation de données dans Data Wrangler

Après avoir configuré un domaine SageMaker AI et lancé Data Wrangler dans un nouvel onglet, vous êtes prêt à importer les données de votre source dans Data Wrangler. Lorsque vous utilisez Data Wrangler pour préparer des données pour Amazon Personalize, vous importez un ensemble de données à la fois. Nous vous recommandons de commencer par un ensemble de données d'interactions entre objets. Vous ne pouvez pas utiliser Data Wrangler pour préparer un ensemble de données Actions ou Action interactions.

Vous commencez sur la page Flux de données. La page doit ressembler à ce qui suit.



Pour commencer à importer des données, sélectionnez Importer des données et spécifiez votre source de données. Data Wrangler prend en charge plus de 40 sources. Il s'agit notamment de AWS services, tels qu'Amazon Redshift, Amazon EMR ou Amazon Athena, et de tiers, tels que Snowflake ou. DataBricks Les différentes sources de données utilisent des procédures différentes pour connecter et importer des données.

Pour obtenir la liste complète des sources disponibles et des instructions détaillées sur l'importation de données, consultez la section Importer dans le manuel Amazon SageMaker AI Developer Guide.

Après avoir importé des données dans Data Wrangler, vous êtes prêt à les transformer. Pour plus d'informations sur la transformation des données, consultezTransformation de données.

Transformation de données

Pour transformer les données dans Data Wrangler, vous devez ajouter une étape de transformation à votre flux de données. Data Wrangler inclut plus de 300 transformations que vous pouvez utiliser pour préparer vos données, y compris une colonne de carte pour la transformation Amazon Personalize. Vous pouvez également utiliser les transformations générales de Data Wrangler pour résoudre des problèmes tels que les valeurs aberrantes, les problèmes de type et les valeurs manquantes.

Une fois que vous avez fini de transformer vos données, vous pouvez les analyser avec Data Wrangler. Ou, si vous avez terminé de préparer vos données dans Data Wrangler, vous pouvez les traiter et les importer dans Amazon Personalize. Pour plus d'informations sur l'analyse des données, consultezGénération de visualisations et d'informations sur les données. Pour plus d'informations sur le traitement et l'importation de données, consultezTraitement des données et importation de celles-ci dans Amazon Personalize.

Rubriques

- Colonnes de mappage pour Amazon Personalize
- Transformations de General Data Wrangler

Colonnes de mappage pour Amazon Personalize

Pour transformer vos données afin qu'elles répondent aux exigences d'Amazon Personalize, vous devez ajouter les colonnes de carte pour la transformation Amazon Personalize et mapper vos colonnes aux champs obligatoires et facultatifs d'Amazon Personalize.

Pour utiliser les colonnes de la carte pour la transformation Amazon Personalize

1. Choisissez + pour votre dernière transformation, puis choisissez Ajouter une transformation. Si vous n'avez pas ajouté de transformation, sélectionnez le signe + pour la transformation des types de données. Data Wrangler ajoute automatiquement cette transformation à votre flux.
2. Choisissez Ajouter une étape.
3. Choisissez Transforms for Amazon Personalize. Les colonnes de carte pour la transformation Amazon Personalize sont sélectionnées par défaut.
4. Utilisez les champs de transformation pour mapper vos données aux attributs Amazon Personalize requis.

    1. Choisissez le type de jeu de données qui correspond à vos données (interactions, éléments ou utilisateurs).
    2. Choisissez votre domaine (ECOMMERCE, VIDEO_ON_DEMAND ou personnalisé). Le domaine que vous choisissez doit correspondre au domaine que vous avez spécifié lors de la création de votre groupe de jeux de données.
    3. Choisissez les colonnes qui correspondent aux champs obligatoires et facultatifs pour Amazon Personalize. Par exemple, pour la colonne Item_ID, choisissez la colonne de vos données qui stocke les informations d'identification uniques pour chacun de vos articles.

        Chaque champ de colonne est filtré par type de données. Seules les colonnes de vos données qui répondent aux exigences relatives aux types de données Amazon Personalize sont disponibles. Si vos données ne sont pas du type requis, vous pouvez utiliser la transformation Parse Value as Type Data Wrangler pour les convertir.

Transformations de General Data Wrangler

Les transformations générales de Data Wrangler suivantes peuvent vous aider à préparer les données pour Amazon Personalize :

- Conversion du type de données : si votre champ ne figure pas parmi les options possibles dans les colonnes de la carte pour la transformation Amazon Personalize, vous devrez peut-être convertir son type de données. Le Data Wrangler transforme Parse Value as Type pour vous aider à convertir vos données. Vous pouvez également utiliser la transformation des types de données que Data Wrangler ajoute par défaut lorsque vous créez un flux. Pour utiliser cette transformation, vous devez sélectionner le type de données dans les listes déroulantes Type, choisir Aperçu, puis choisir Mettre à jour.

Pour plus d'informations sur les types de données requis pour les champs, consultez la section relative à votre domaine et au type de jeu de données dansCréation de fichiers JSON pour les schémas Amazon Personalize.

- Gestion des valeurs manquantes et des valeurs aberrantes : si vous générez des informations sur les valeurs manquantes ou les valeurs aberrantes, vous pouvez utiliser le Data Wrangler qui transforme les transformations Handle Outliers et Handle Missing Values pour résoudre ces problèmes.
- Transformations personnalisées : avec Data Wrangler, vous pouvez créer vos propres transformations avec Python (User-Defined Function) PySpark, Pandas ou PySpark (SQL). Vous pouvez utiliser une transformation personnalisée pour effectuer des tâches telles que la suppression de colonnes dupliquées ou le regroupement par colonnes. Pour plus d'informations, consultez la section Custom Transforms du manuel Amazon SageMaker AI Developer Guide.

Génération de visualisations et d'informations sur les données

Après avoir importé vos données dans Data Wrangler, vous pouvez les utiliser pour générer des visualisations et des informations sur les données.

- Visualisations : Data Wrangler peut générer différents types de graphiques, tels que des histogrammes et des diagrammes de points. Par exemple, vous pouvez générer un histogramme pour identifier les valeurs aberrantes dans vos données.
- Informations sur les données : vous pouvez utiliser un rapport sur la qualité et les informations des données pour Amazon Personalize pour en savoir plus sur vos données grâce à des informations sur les données et à des statistiques sur les colonnes et les lignes. Ce rapport peut vous indiquer si vous rencontrez des problèmes de type dans vos données. Et vous pouvez découvrir les mesures que vous pouvez prendre pour améliorer vos données. Ces actions peuvent vous aider à répondre aux besoins en ressources d'Amazon Personalize, tels que les exigences en matière de formation des modèles, ou elles peuvent mener à de meilleures recommandations.

Après avoir découvert vos données grâce à des visualisations et à des informations, vous pouvez utiliser ces informations pour appliquer des transformations supplémentaires afin d'améliorer vos données. Ou, si vous avez terminé de préparer vos données, vous pouvez les traiter et les importer dans Amazon Personalize. Pour plus d'informations sur la transformation de vos données, consultezTransformation de données. Pour plus d'informations sur le traitement et l'importation de données, consultezTraitement des données et importation de celles-ci dans Amazon Personalize.

Génération de visualisations

Vous pouvez utiliser Data Wrangler pour créer différents types de graphiques, tels que des histogrammes et des diagrammes de points. Par exemple, vous pouvez générer un histogramme pour identifier les valeurs aberrantes dans vos données. Pour générer une visualisation des données, vous ajoutez une étape d'analyse à votre flux et, dans Type d'analyse, choisissez la visualisation que vous souhaitez créer.

Pour plus d'informations sur la création de visualisations dans Data Wrangler, consultez Analyser et visualiser dans le manuel Amazon SageMaker AI Developer Guide.

Générer des informations sur les données

Vous pouvez utiliser Data Wrangler pour générer un rapport sur la qualité et les informations des données pour Amazon Personalize spécifique à votre type de jeu de données. Avant de générer le rapport, nous vous recommandons de transformer vos données pour répondre aux exigences d'Amazon Personalize. Cela permettra d'obtenir des informations plus pertinentes. Pour plus d'informations, consultez Transformation de données.

Rubriques

- Contenu du rapport
- Génération du rapport

Contenu du rapport

Le rapport sur la qualité et les informations des données pour Amazon Personalize comprend les sections suivantes :

- Résumé : Le résumé du rapport inclut les statistiques du jeu de données et les avertissements prioritaires :
  - Statistiques du jeu de données : il s'agit notamment des statistiques spécifiques d'Amazon Personalize, telles que le nombre d'utilisateurs uniques dans vos données d'interactions, et des statistiques générales, telles que le nombre de valeurs manquantes ou aberrantes.
  - Avertissements prioritaires : il s'agit des informations spécifiques à Amazon Personalize qui ont le plus d'impact sur la formation ou les recommandations. Chaque avertissement inclut une mesure recommandée que vous pouvez prendre pour résoudre le problème.
- Lignes dupliquées et lignes incomplètes : ces sections contiennent des informations sur les lignes contenant des valeurs manquantes et les lignes dupliquées dans vos données.

- Résumé des fonctionnalités : Cette section inclut le type de données pour chaque colonne, les informations de données non valides ou manquantes et le nombre d'avertissements.
- Détails des fonctionnalités : Cette section comprend des sous-sections contenant des informations détaillées pour chacune de vos colonnes de données. Chaque sous-section inclut des statistiques pour la colonne, telles que le nombre de valeurs catégorielles et les informations sur les valeurs manquantes. Et chaque sous-section inclut des informations spécifiques d'Amazon Personalize et des actions recommandées pour les colonnes de données. Par exemple, un aperçu peut indiquer qu'une colonne comporte plus de 30 catégories possibles.

Problèmes liés aux types de données

Le rapport identifie les colonnes dont le type de données n'est pas correct et précise le type requis. Pour obtenir des informations relatives à ces fonctionnalités, vous devez convertir le type de données de la colonne et générer à nouveau le rapport. Pour convertir le type, vous pouvez utiliser le Data Wrangler pour transformer Parse Value as Type.

Amazon Personalize : informations

Les informations d'Amazon Personalize incluent une constatation et une action suggérée. L'action est facultative. Par exemple, le rapport peut inclure un aperçu et une action liés au nombre de catégories pour une colonne de données catégorielles. Si vous pensez que la chronique n'est pas catégorique, vous pouvez ignorer cette information et ne rien faire.

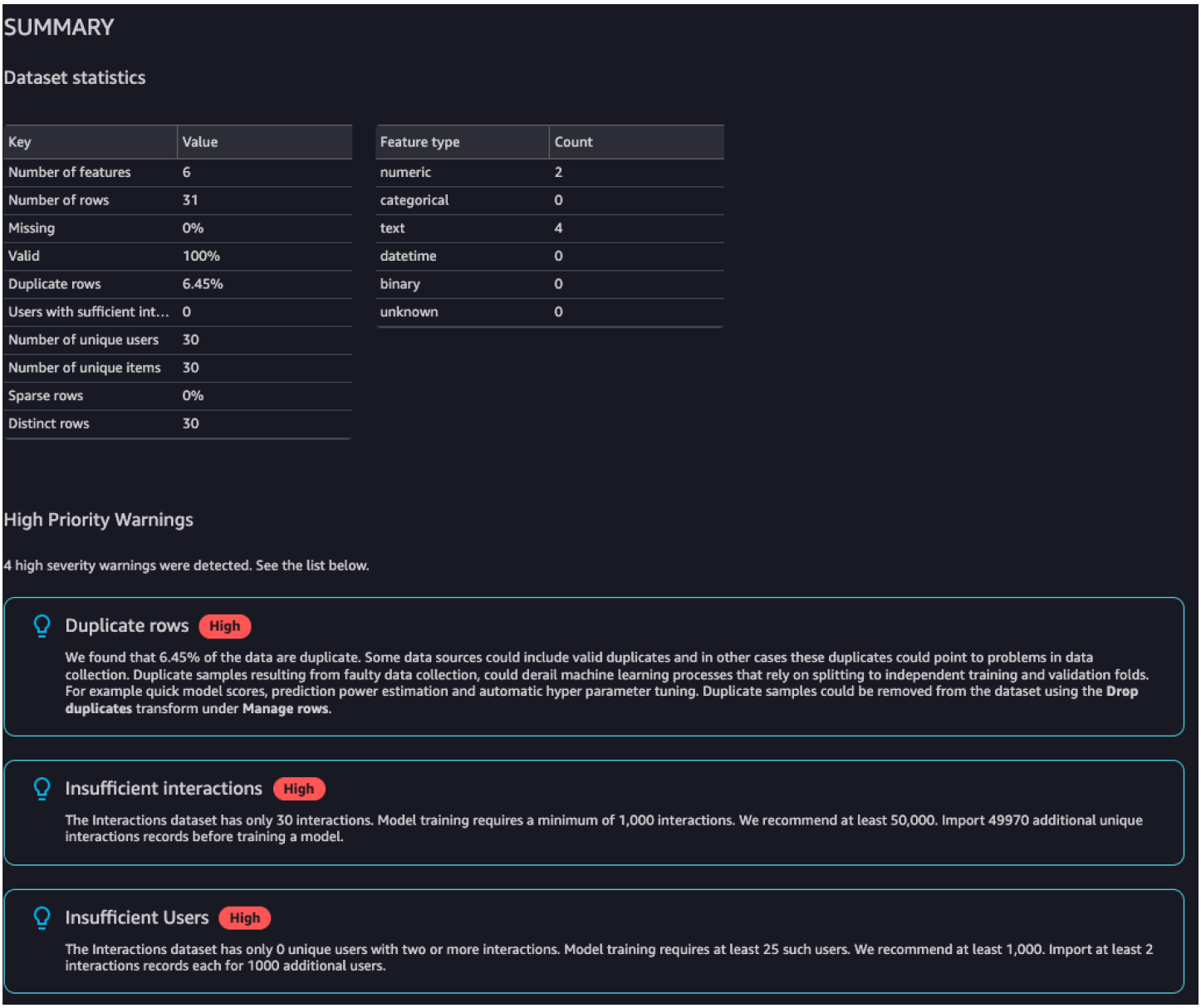
À l'exception de légères différences de formulation, les informations spécifiques à Amazon Personalize sont les mêmes que les informations relatives à un seul ensemble de données que vous pouvez générer lorsque vous analysez vos données avec Amazon Personalize. Par exemple, le rapport d'analyse de Data Wrangler inclut des informations telles que « Le jeu de données sur les interactions entre éléments ne compte que X utilisateurs uniques avec deux interactions ou plus ». Mais il n'inclut pas d'informations telles que « X % des éléments du jeu de données Items n'ont aucune interaction dans le jeu de données des interactions Item ».

Pour obtenir une liste des informations spécifiques possibles à Amazon Personalize, consultez les informations qui ne font pas référence à Informations sur les données plusieurs ensembles de données.

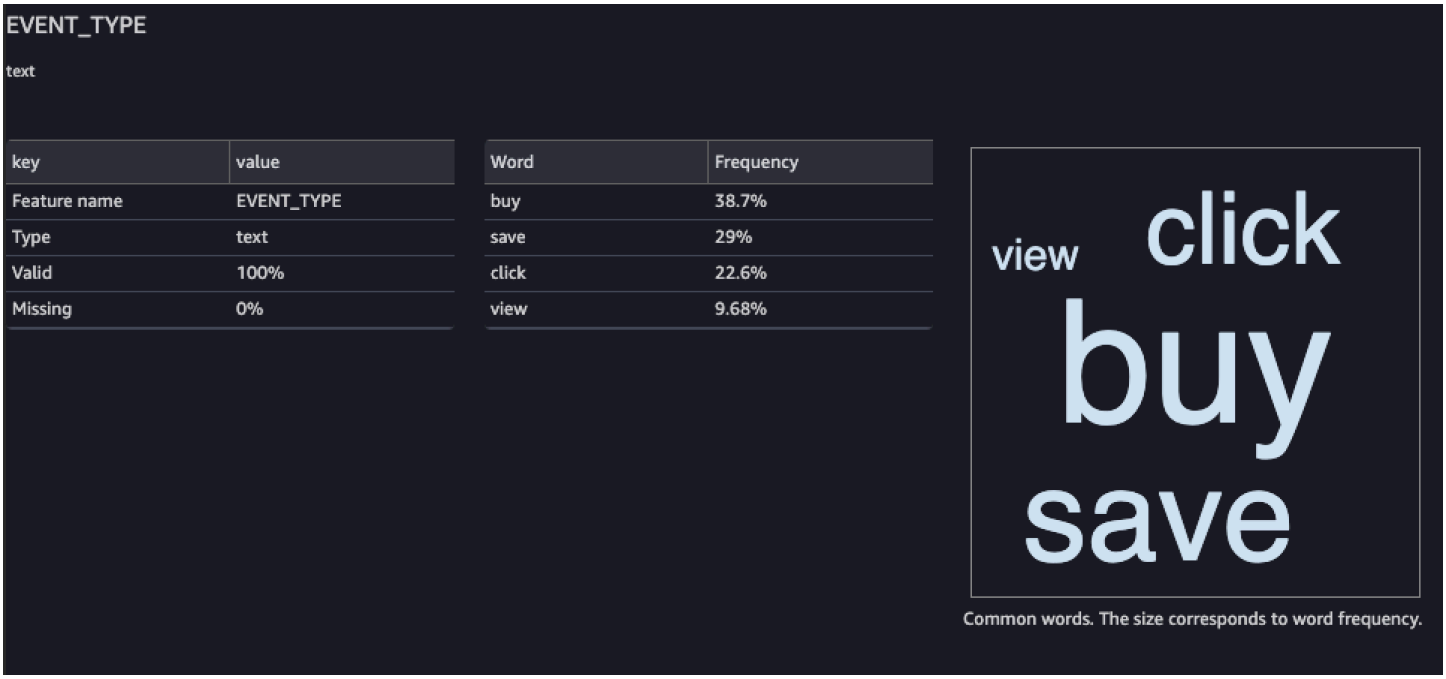
Exemples de rapports

L'apparence du rapport Amazon Personalize est identique à celle du rapport d'analyse générale de Data Wrangler. Pour des exemples du rapport d'analyse générale, consultez Get Insights

On Data and Data Quality dans le manuel Amazon SageMaker AI Developer Guide. L'exemple suivant montre comment fonctionne la section récapitulative d'un rapport pour un ensemble de données d'interactions entre articles. Il inclut des statistiques sur les ensembles de données et certains avertissements possibles concernant les interactions entre objets et ensembles de données présentant une priorité élevée.



L'exemple suivant montre comment la section des détails des fonctionnalités d'une colonne EVENT_TYPE d'un ensemble de données d'interactions entre éléments peut apparaître dans un rapport.



Génération du rapport

Pour générer le rapport sur la qualité et les informations des données pour Amazon Personalize, vous sélectionnez Obtenir des informations sur les données pour votre transformation et créer une analyse.

Pour générer un rapport sur la qualité et les informations des données pour Amazon Personalize

1. Choisissez l'option + pour la transformation que vous analysez. Si vous n'avez pas ajouté de transformation, sélectionnez le signe + pour la transformation des types de données. Data Wrangler ajoute automatiquement cette transformation à votre flux.
2. Choisissez Obtenir des informations sur les données. Le panneau Créer une analyse s'affiche.
3. Pour le type d'analyse, choisissez Data Quality and Insights Report for Amazon Personalize.
4. Pour le type de jeu de données, choisissez le type de jeu de données Amazon Personalize que vous analysez.
5. Choisissez éventuellement Exécuter sur des données complètes. Par défaut, Data Wrangler génère des informations uniquement sur un échantillon de vos données.
6. Cliquez sur Créer. Lorsque l'analyse est terminée, le rapport apparaît.

Traitement des données et importation de celles-ci dans Amazon Personalize

Lorsque vous avez terminé d'analyser et de transformer vos données, vous êtes prêt à les traiter et à les importer dans Amazon Personalize.

- Traitement des données : le traitement des données applique votre transformation à l'ensemble de votre ensemble de données et la renvoie vers une destination que vous spécifiez. Dans ce cas, vous spécifiez un compartiment Amazon S3.
- Importation de données dans Amazon Personalize — Pour importer des données traitées dans Amazon Personalize, vous devez exécuter un bloc-notes Jupyter fourni dans SageMaker AI Studio Classic. Ce bloc-notes crée vos ensembles de données Amazon Personalize et y importe vos données.

Traitement des données

Avant d'importer des données dans Amazon Personalize, vous devez appliquer votre transformation à l'ensemble de votre ensemble de données et la transférer dans un compartiment Amazon S3. Pour ce faire, vous créez un nœud de destination dont la destination est définie sur un compartiment Amazon S3, puis vous lancez une tâche de traitement pour la transformation.

Pour obtenir des instructions détaillées sur la spécification d'une destination et le lancement d'une tâche de traitement, consultez Lancer des tâches de traitement en quelques clics à l'aide d'Amazon SageMaker AI Data Wrangler. Lorsque vous ajoutez une destination, choisissez Amazon S3. Vous utiliserez cet emplacement lorsque vous importerez les données traitées dans Amazon Personalize.

Lorsque vous avez terminé de traiter vos données, vous êtes prêt à les importer depuis le compartiment Amazon S3 dans Amazon Personalize.

Importation de données dans Amazon Personalize

Après avoir traité vos données, vous êtes prêt à les importer dans Amazon Personalize. Pour importer des données traitées dans Amazon Personalize, vous devez exécuter un bloc-notes Jupyter fourni dans SageMaker AI Studio Classic. Ce bloc-notes crée vos ensembles de données Amazon Personalize et y importe vos données.

Pour importer des données traitées dans Amazon Personalize

1. Pour la transformation que vous souhaitez exporter, choisissez Exporter vers et Amazon Personalize (via Jupyter Notebook).
2. Modifiez le bloc-notes pour spécifier le compartiment Amazon S3 que vous avez utilisé comme destination des données pour la tâche de traitement. Spécifiez éventuellement le domaine de votre groupe de jeux de données. Par défaut, le bloc-notes crée un groupe de jeux de données personnalisé.
3. Passez en revue les cellules du bloc-notes qui créent le schéma. Vérifiez que les champs du schéma possèdent les types et les attributs attendus avant d'exécuter la cellule.

   - Vérifiez que les champs prenant en charge les données nulles `null` figurent dans la liste des types. L'exemple suivant montre comment ajouter `null` pour un champ.

     ```
     {
       "name": "GENDER",
       "type": [
         "null",
         "string"
       ],
       "categorical": true
     }
     ```

   - Vérifiez que l'attribut catégoriel est défini sur true pour les champs catégoriels. L'exemple suivant montre comment marquer un champ comme catégorique.

     ```
     {
               "name": "SUBSCRIPTION_MODEL",
               "type": "string",
               "categorical": true
     }
     ```

   - Vérifiez que l'attribut textuel des champs textuels est défini sur true. L'exemple suivant montre comment marquer un champ comme textuel.

     ```
     {
           "name": "DESCRIPTION",
           "type": [
             "null",
             "string"
           ],
     ```

```
        "textual": true
}
```

4. Exécutez le bloc-notes pour créer un schéma, créer un ensemble de données et importer vos données dans le jeu de données Amazon Personalize. Vous exécutez le bloc-notes comme vous le feriez avec un bloc-notes en dehors d' SageMaker AI Studio Classic. Pour plus d'informations sur l'exécution des blocs-notes Jupyter, consultez Running Code. Pour plus d'informations sur les blocs-notes dans SageMaker AI Studio Classic, consultez la section Utiliser les blocs-notes Amazon SageMaker AI dans le manuel Amazon SageMaker AI Developer Guide.

   Une fois le bloc-notes terminé, si vous avez importé des données d'interactions, vous êtes prêt à créer des recommandations ou des ressources personnalisées. Vous pouvez également répéter le processus avec un jeu de données d'éléments ou un ensemble de données d'utilisateurs.

   - Pour plus d'informations sur la création de recommandeurs de domaine, consultezRecommandeurs de domaines dans Amazon Personalize.
   - Pour plus d'informations sur la création et le déploiement de ressources personnalisées, consultezRessources personnalisées pour la formation et le déploiement des modèles Amazon Personalize.

Importation d'enregistrements individuels dans un ensemble de données Amazon Personalize

Une fois que vous avez terminéCréation d'un schéma et d'un jeu de données, vous pouvez importer des enregistrements individuels, y compris les interactions entre les éléments, les utilisateurs, les éléments, les actions ou les interactions entre les actions dans un ensemble de données existant. L'importation de données individuelles vous permet d'ajouter de petits lots d'enregistrements à vos ensembles de données Amazon Personalize à mesure que votre catalogue s'agrandit. Vous pouvez importer jusqu'à 10 enregistrements par opération d'importation individuelle.

Si vous importez un article, un utilisateur ou une action avec le même identifiant qu'un enregistrement figurant déjà dans votre ensemble de données, Amazon Personalize le remplace par le nouvel enregistrement. Si vous enregistrez deux événements d'interaction entre un article ou une action avec exactement le même horodatage et les mêmes propriétés, Amazon Personalize ne conserve qu'un seul des événements.

Si vous utilisez Apache Kafka, vous pouvez utiliser le connecteur Kafka pour Amazon Personalize afin de diffuser des données en temps réel vers Amazon Personalize. Pour plus d'informations,

consultez Kafka Connector pour Amazon Personalize dans le référentiel personalize-kafka-connectorGithub.

Si vous disposez d'un grand nombre d'enregistrements historiques, nous vous recommandons d'importer d'abord les données en bloc, puis de les importer individuellement si nécessaire. Consultez Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données.

Mises à jour des filtres pour les importations d'enregistrements individuels

Amazon Personalize met à jour tous les filtres que vous avez créés dans le groupe de données avec vos nouvelles données d'interaction, d'article et d'utilisateur dans les 20 minutes suivant la dernière importation individuelle. Cette mise à jour permet à vos campagnes d'utiliser vos données les plus récentes pour filtrer les recommandations destinées à vos utilisateurs.

Si vous avez déjà créé un recommandateur ou déployé une version de solution personnalisée dans le cadre d'une campagne, l'influence des nouveaux enregistrements individuels sur les recommandations dépend du cas d'utilisation du domaine ou de la recette que vous utilisez. Pour de plus amples informations, veuillez consulter Mise à jour des données dans les ensembles de données après l'entraînement.

Rubriques

- Importer des interactions individuellement
- Importer des utilisateurs individuellement
- Importer des articles individuellement
- Importer des actions individuellement

Importer des interactions individuellement

Une fois que vous Création d'un schéma et d'un jeu de données avez créé un jeu de données d'interactions entre articles, vous pouvez importer individuellement un ou plusieurs nouveaux événements dans le jeu de données. Pour importer des événements d'interaction individuellement, vous créez un outil de suivi des événements, puis vous importez un ou plusieurs événements dans votre jeu de données d'interactions avec les articles. Vous pouvez importer des événements d'interaction individuels historiques à l'aide de la console Amazon Personalize, ou importer des événements historiques ou en temps réel à l'aide du AWS Command Line Interface (AWS CLI) ou du AWS SDKs.

Cette section contient des informations sur l'importation d'événements avec la console Amazon Personalize. Nous vous recommandons d'utiliser la console Amazon Personalize pour importer uniquement des événements historiques. Pour plus d'informations sur l'utilisation du AWS CLI ou AWS SDKs pour enregistrer des événements en temps réel, consultezEnregistrement des événements en temps réel pour influencer les recommandations.

Pour plus d'informations sur la manière dont Amazon Personalize met à jour les filtres pour les nouveaux enregistrements et sur la manière dont les nouveaux enregistrements influencent les recommandations, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize.

Rubriques

- Création d'un outil de suivi des événements (console)
- Importation d'événements individuellement (console)

Création d'un outil de suivi des événements (console)

> **Note**
>
> Si vous avez créé un outil de suivi des événements, vous pouvez passer àImportation d'événements individuellement (console).

Avant de pouvoir importer un événement dans un jeu de données Interactions, vous devez créer un outil de suivi des événements pour le groupe de jeux de données.

Pour créer un outil de suivi des événements (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant le jeu de données d'interactions entre éléments dans lequel vous souhaitez importer des événements.
3. Sur le tableau de bord du groupe de jeux de données, dans Installer le SDK d'ingestion d'événements, sélectionnez Start.
4. Sur la page Configurer le tracker, dans Configurations du tracker, pour Nom du tracker, saisissez un nom pour le suivi des événements, puis choisissez Next.

5. La page Installer le SDK indique l'ID de suivi du nouveau système de suivi des événements et les instructions relatives à l'utilisation AWS Amplify ou AWS Lambda à la diffusion des données relatives aux événements.

   Vous pouvez ignorer ces informations car vous utilisez la console Amazon Personalize pour télécharger les données des événements. Si vous souhaitez diffuser des données d'événements AWS Lambda en utilisant AWS Amplify ou à l'avenir, vous pouvez consulter ces informations en choisissant le suivi des événements sur la page Suivi des événements.
6. Choisissez Finish (Terminer). Vous pouvez désormais importer des événements avec la console (voir Importation d'événements individuellement (console) ou enregistrer des événements en temps réel à l'aide de l'`PutEvents`opération (voirEnregistrement des événements en temps réel pour influencer les recommandations)).

Importation d'événements individuellement (console)

Après avoir créé un outil de suivi des événements, vous pouvez importer les événements individuellement dans un jeu de données d'interactions entre articles. Cette procédure part du principe que vous avez déjà créé un jeu de données d'interactions entre les éléments. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Pour importer des événements individuellement (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant le jeu de données d'interactions entre éléments dans lequel vous souhaitez importer des événements.
3. Dans le volet de navigation, choisissez des ensembles de données.
4. Sur la page Ensembles de données, choisissez le jeu de données Interactions.
5. En haut à droite de la page des détails de l'ensemble de données, choisissez Modifier l'ensemble de données, puis Créer un enregistrement.
6. Dans la page Créer un ou plusieurs enregistrements d'interaction entre l'utilisateur et l'élément, pour la saisie d'un enregistrement, entrez les détails de l'événement au format JSON. Les noms et valeurs des champs de l'événement doivent correspondre au schéma que vous avez utilisé lorsque vous avez créé le jeu de données d'interactions entre éléments. Amazon Personalize

fournit un modèle JSON avec les noms de champs et les types de données issus de ce schéma. Vous pouvez importer jusqu'à 10 événements à la fois.

7. Choisissez Créer un ou plusieurs enregistrements. Dans Response, le résultat de l'importation est répertorié et un message de réussite ou d'échec s'affiche.

Importer des utilisateurs individuellement

Une fois que vous Création d'un schéma et d'un jeu de données avez créé un jeu de données Utilisateurs, vous pouvez importer individuellement un ou plusieurs nouveaux utilisateurs dans le jeu de données. L'importation individuelle d'utilisateurs vous permet de maintenir votre jeu de données Utilisateurs à jour grâce à des importations par petits lots à mesure que votre catalogue s'agrandit. Vous pouvez importer jusqu'à 10 utilisateurs à la fois. Si vous avez un grand nombre de nouveaux utilisateurs, nous vous recommandons d'importer d'abord les données en masse, puis de les importer individuellement si nécessaire. Consultez Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données.

Vous pouvez utiliser la console Amazon Personalize, le AWS Command Line Interface (AWS CLI), ou AWS SDKs pour importer des utilisateurs. Si vous importez un utilisateur `userId` identique à un utilisateur figurant déjà dans votre ensemble de données Utilisateurs, Amazon Personalize le remplace par le nouveau. Vous pouvez importer jusqu'à 10 utilisateurs à la fois.

Pour plus d'informations sur la manière dont Amazon Personalize met à jour les filtres pour les nouveaux enregistrements et sur la manière dont les nouveaux enregistrements influencent les recommandations, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize.

Rubriques

- Importer des utilisateurs individuellement (console)
- Importer des utilisateurs individuellement (AWS CLI)
- Importer des utilisateurs individuellement (AWS SDKs)

Importer des utilisateurs individuellement (console)

Vous pouvez importer jusqu'à 10 utilisateurs à la fois. Cette procédure suppose que vous avez déjà créé un jeu de données Users. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Pour importer des utilisateurs individuellement (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant le jeu de données Utilisateurs dans lequel vous souhaitez importer l'utilisateur.
3. Dans le volet de navigation, sélectionnez Datasets.
4. Sur la page Ensembles de données, choisissez le jeu de données Utilisateurs.
5. Sur la page des détails du jeu de données, en haut à droite, choisissez Modifier le jeu de données, puis Créer un enregistrement.
6. Sur la page Créer un ou plusieurs enregistrements utilisateur, pour la saisie des enregistrements, entrez les détails de l'utilisateur au format JSON. Les noms et valeurs des champs de l'utilisateur doivent correspondre au schéma que vous avez utilisé lors de la création du jeu de données Utilisateurs. Amazon Personalize fournit un modèle JSON avec les noms de champs et les types de données issus de ce schéma.
7. Choisissez Créer un ou plusieurs enregistrements. Dans Réponse, le résultat de l'importation est répertorié et un message de réussite ou d'échec s'affiche.

Importer des utilisateurs individuellement (AWS CLI)

Ajoutez un ou plusieurs utilisateurs à votre jeu de données Users avec l'PutUsersopération. Vous pouvez importer jusqu'à 10 utilisateurs en un seul `PutUsers` appel. Cette section part du principe que vous avez déjà créé un ensemble de données Utilisateurs. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Utilisez la `put-users` commande suivante pour ajouter un ou plusieurs utilisateurs avec le AWS CLI. `dataset arn`Remplacez-le par le Amazon Resource Name (ARN) de votre ensemble de données et `user Id` par l'ID de l'utilisateur. Si un utilisateur correspondant `userId` figure déjà dans votre ensemble de données Users, Amazon Personalize le remplace par le nouveau.

En `properties` effet, pour chaque champ de votre jeu de données Utilisateurs, remplacez le `propertyName` par le nom du champ de votre schéma en cas de chameau. Par exemple, GENDER serait `gender` et MEMBERSHIP_TYPE serait. `membershipType` Remplacez `user data` par les données de l'utilisateur. Pour les données de chaîne catégorielles, afin d'inclure plusieurs catégories pour une seule propriété, séparez chaque catégorie par un tube (`|`). Par exemple `\"Premium Class|Legacy Member\"`.

```
aws personalize-events put-users \
  --dataset-arn dataset arn \
  --users '[{
      "userId": "user Id",
      "properties": "{\"propertyName\": "\user data\"}"
    },
    {
      "userId": "user Id",
      "properties": "{\"propertyName\": "\user data\"}"
    }]'
```

Importer des utilisateurs individuellement (AWS SDKs)

Ajoutez un ou plusieurs utilisateurs à votre jeu de données Users avec l'PutUsersopération. Si un utilisateur correspondant `userId` figure déjà dans votre ensemble de données Users, Amazon Personalize le remplace par le nouveau. Vous pouvez importer jusqu'à 10 utilisateurs en un seul `PutUsers` appel. Cette section part du principe que vous avez déjà créé un ensemble de données Utilisateurs. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Le code suivant montre comment ajouter un ou plusieurs utilisateurs à votre jeu de données Utilisateurs. Pour chaque paramètre de nom de propriété, transmettez le nom du champ de votre schéma en cas de chameau. Par exemple, GENDER serait `gender` et MEMBERSHIP_TYPE serait. `membershipType` Pour chaque paramètre de valeur de propriété, transmettez les données à l'utilisateur.

Pour les données de chaîne catégorielles, pour inclure plusieurs catégories pour une seule propriété, séparez chaque catégorie par un tube (`|`). Par exemple `"Premium class|Legacy Member"`.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_users(
    datasetArn = 'dataset arn',
    users = [{
      'userId': 'user ID',
      'properties': "{\"propertyName\": \"user data\"}"
      },
```

```
      {
      'userId': 'user ID',
      'properties': "{\"propertyName\": \"user data\"}"
      }]
)
```

SDK for Java 2.x

```
public static int putUsers(PersonalizeEventsClient personalizeEventsClient,
                           String datasetArn,
                           String user1Id,
                           String user1PropertyName,
                           String user1PropertyValue,
                           String user2Id,
                           String user2PropertyName,
                           String user2PropertyValue) {

    int responseCode = 0;
    ArrayList<User> users = new ArrayList<>();

    try {
        User user1 = User.builder()
          .userId(user1Id)
          .properties(String.format("{\"%1$s\": \"%2$s\"}", user1PropertyName,
 user1PropertyValue))
          .build();

        users.add(user1);

        User user2 = User.builder()
          .userId(user2Id)
          .properties(String.format("{\"%1$s\": \"%2$s\"}", user2PropertyName,
 user2PropertyValue))
          .build();

        users.add(user2);

        PutUsersRequest putUsersRequest = PutUsersRequest.builder()
          .datasetArn(datasetArn)
          .build();

        responseCode =
 personalizeEventsClient.putUsers(putUsersRequest).sdkHttpResponse().statusCode();
```

```
            System.out.println("Response code: " + responseCode);
            return responseCode;

        } catch (PersonalizeEventsException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return responseCode;
    }
```

SDK for JavaScript v3

```
import {
  PutUsersCommand,
  PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
  region: "REGION",
});

// set the put users parameters
var putUsersParam = {
  datasetArn:
    "DATASET ARN",
  users: [
    {
      userId: "userId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
    {
      userId: "userId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
  ],
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutUsersCommand(putUsersParam)
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
```

```
    console.log("Error", err);
  }
};
run();
```

Importer des articles individuellement

Une fois que vous Création d'un schéma et d'un jeu de données avez créé un jeu de données d'éléments, vous pouvez importer individuellement un ou plusieurs nouveaux éléments dans le jeu de données. L'importation individuelle d'articles vous permet de maintenir votre jeu de données d'articles à jour grâce à des importations par petits lots à mesure que votre catalogue s'agrandit. Vous pouvez importer jusqu'à 10 articles à la fois. Si vous avez un grand nombre de nouveaux articles, nous vous recommandons d'importer d'abord les données en bloc, puis d'importer les données des articles individuellement si nécessaire. Consultez Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données.

Vous pouvez utiliser la console Amazon Personalize, le AWS Command Line Interface (AWS CLI), ou AWS SDKs pour importer des articles. Si vous importez un article `itemId` identique à un article figurant déjà dans votre ensemble de données d'articles, Amazon Personalize le remplace par le nouvel article.

Pour plus d'informations sur la manière dont Amazon Personalize met à jour les filtres pour les nouveaux enregistrements et sur la manière dont les nouveaux enregistrements influencent les recommandations, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize.

Rubriques

- Importer des éléments individuellement (console)
- Importer des articles individuellement (AWS CLI)
- Importer des articles individuellement (AWS SDKs)

Importer des éléments individuellement (console)

Vous pouvez importer jusqu'à 10 éléments à la fois dans un jeu de données d'articles. Cette procédure suppose que vous avez déjà créé un jeu de données Items. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Pour importer des articles individuellement (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant le jeu de données Éléments dans lequel vous souhaitez importer les éléments.
3. Dans le volet de navigation, sélectionnez Datasets.
4. Sur la page Ensembles de données, choisissez le jeu de données Items.
5. En haut à droite de la page des détails de l'ensemble de données, choisissez Modifier le jeu de données, puis Créer un enregistrement.
6. Dans la page Créer un ou plusieurs enregistrements d'articles, pour la saisie d'un enregistrement, entrez les détails de l'article au format JSON. Les noms et valeurs des champs de l'élément doivent correspondre au schéma que vous avez utilisé lors de la création du jeu de données Items. Amazon Personalize fournit un modèle JSON avec les noms de champs et les types de données issus de ce schéma.
7. Choisissez Créer un ou plusieurs enregistrements. Dans Réponse, le résultat de l'importation est répertorié et un message de réussite ou d'échec s'affiche.

Importer des articles individuellement (AWS CLI)

Ajoutez un ou plusieurs éléments à votre jeu de données Items à l'aide de PutItems cette opération. Vous pouvez importer jusqu'à 10 articles en un seul `PutItems` appel. Cette section part du principe que vous avez déjà créé un jeu de données Items. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Utilisez la `put-items` commande suivante pour ajouter un ou plusieurs éléments avec le AWS CLI. `dataset arn`Remplacez-le par le Amazon Resource Name (ARN) de votre ensemble de données et `item Id` par l'ID de l'élément. Si un article identique `itemId` figure déjà dans votre ensemble de données Articles, Amazon Personalize le remplace par le nouveau.

En effet`properties`, pour chaque champ de votre jeu de données Items, remplacez le `propertyName` par le nom du champ indiqué dans votre schéma en cas de chameau. Par exemple, GENRES serait `genres` et CREATION_TIMESTAMP serait CreationTimestamp. Remplacez `item data` par les données de l'article. `CREATION_TIMESTAMP`les données doivent être au format Epoch Time Unix et exprimées en secondes. Pour les données de chaîne catégorielles, afin d'inclure

plusieurs catégories pour une seule propriété, séparez chaque catégorie par un tube (|). Par exemple \"Horror|Action\".

```
aws personalize-events put-items \
  --dataset-arn dataset arn \
  --items '[{
      "itemId": "item Id",
      "properties": "{\"propertyName\": "\item data\"}"
    },
    {
      "itemId": "item Id",
      "properties": "{\"propertyName\": "\item data\"}"
    }]'
```

Importer des articles individuellement (AWS SDKs)

Ajoutez un ou plusieurs éléments à votre jeu de données Items à l'aide de PutItems cette opération. Vous pouvez importer jusqu'à 10 articles en un seul `PutItems` appel. Si un article identique `itemId` figure déjà dans votre ensemble de données Articles, Amazon Personalize le remplace par le nouveau. Cette section part du principe que vous avez déjà créé un jeu de données Items. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Le code suivant montre comment ajouter un ou plusieurs éléments à votre jeu de données Items. Pour chaque paramètre de nom de propriété, transmettez le nom du champ de votre schéma en cas de chameau. Par exemple, GENRES serait `genres` et CREATION_TIMESTAMP serait. `creationTimestamp` Pour chaque paramètre de valeur de propriété, transmettez les données de l'élément. `CREATION_TIMESTAMP`Les données doivent être au format Epoch Time Unix et exprimées en secondes.

Pour les données de chaîne catégorielles, afin d'inclure plusieurs catégories pour une seule propriété, séparez chaque catégorie par un tube (|). Par exemple `"Horror|Action"`.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_items(
```

```
    datasetArn = 'dataset arn',
    items = [{
      'itemId': 'item ID',
      'properties': "{\"propertyName\": \"item data\"}"
      },
      {
      'itemId': 'item ID',
      'properties': "{\"propertyName\": \"item data\"}"
      }]
)
```

SDK for Java 2.x

```
public static int putItems(PersonalizeEventsClient personalizeEventsClient,
                           String datasetArn,
                           String item1Id,
                           String item1PropertyName,
                           String item1PropertyValue,
                           String item2Id,
                           String item2PropertyName,
                           String item2PropertyValue) {

    int responseCode = 0;
    ArrayList<Item> items = new ArrayList<>();

    try {
        Item item1 = Item.builder()
                .itemId(item1Id)
                .properties(String.format("{\"%1$s\": \"%2$s\"}",
                        item1PropertyName, item1PropertyValue))
                .build();

        items.add(item1);

        Item item2 = Item.builder()
                .itemId(item2Id)
                .properties(String.format("{\"%1$s\": \"%2$s\"}",
                        item2PropertyName, item2PropertyValue))
                .build();

        items.add(item2);

        PutItemsRequest putItemsRequest = PutItemsRequest.builder()
```

```
                .datasetArn(datasetArn)
                .items(items)
                .build();

        responseCode =
 personalizeEventsClient.putItems(putItemsRequest).sdkHttpResponse().statusCode();
        System.out.println("Response code: " + responseCode);
        return responseCode;

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return responseCode;
    }
```

SDK for JavaScript v3

```
import {
  PutItemsCommand,
  PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
  region: "REGION",
});

// set the put items parameters
var putItemsParam = {
  datasetArn:
    "DATASET ARN",
  items: [
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
  ],
};
export const run = async () => {
  try {
```

```
    const response = await personalizeEventsClient.send(
      new PutItemsCommand(putItemsParam)
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Importer des actions individuellement

Une fois que vous Création d'un schéma et d'un jeu de données avez créé un jeu de données Actions, vous pouvez importer individuellement une ou plusieurs nouvelles actions dans le jeu de données. Lorsque vous importez des actions individuellement, vous maintenez votre jeu de données Actions à jour grâce à des importations par petits lots à mesure que votre catalogue s'agrandit. Vous pouvez importer jusqu'à 10 actions à la fois. Si vous avez un grand nombre de nouvelles actions, nous vous recommandons d'importer d'abord les données en bloc, puis d'importer les données d'action individuellement si nécessaire. Consultez Importation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données.

Vous pouvez utiliser la console Amazon Personalize, le AWS Command Line Interface (AWS CLI), ou AWS SDKs pour importer des actions. Si vous importez une action `actionId` identique à une action déjà présente dans votre ensemble de données Actions, Amazon Personalize la remplace par la nouvelle action.

Pour plus d'informations sur l'influence des nouveaux enregistrements sur les recommandations, voirMise à jour des données dans les ensembles de données après l'entraînement.

Rubriques

- Importer des actions individuellement (console)
- Importer des actions individuellement (AWS CLI)
- Importer des actions individuellement (AWS SDKs)

Importer des actions individuellement (console)

Vous pouvez importer jusqu'à 10 actions à la fois dans un jeu de données Actions. Cette section part du principe que vous avez déjà créé un jeu de données Actions. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Pour importer des actions individuellement (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant le jeu de données Actions auquel vous souhaitez ajouter.
3. Dans le volet de navigation, sélectionnez Datasets.
4. Sur la page Ensembles de données, choisissez le jeu de données Actions.
5. En haut à droite de la page des détails de l'ensemble de données, choisissez Modifier le jeu de données, puis Créer un enregistrement.
6. Dans la page Créer un ou plusieurs enregistrements d'action, pour la saisie d'un enregistrement, entrez les détails de l'action au format JSON. Les noms et valeurs des champs de l'action doivent correspondre au schéma que vous avez utilisé lors de la création du jeu de données Actions. Amazon Personalize fournit un modèle JSON avec les noms de champs et les types de données issus de ce schéma.
7. Choisissez Créer un ou plusieurs enregistrements. Dans Réponse, le résultat de l'importation est répertorié et un message de réussite ou d'échec s'affiche.

Importer des actions individuellement (AWS CLI)

Ajoutez une ou plusieurs actions à votre jeu de données Actions à l'aide de l'opération `PutActions` d'API. Vous pouvez importer jusqu'à 10 actions à la fois. Cette section part du principe que vous avez déjà créé un jeu de données Actions. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Utilisez la `put-actions` commande suivante pour ajouter une ou plusieurs actions avec le AWS CLI. `dataset arn`Remplacez-le par le Amazon Resource Name (ARN) de votre ensemble de données et `actionId` par l'ID de l'action. Si une action identique `actionId` figure déjà dans votre jeu de données Actions, Amazon Personalize la remplace par la nouvelle.

En `properties` effet, pour chaque champ de votre jeu de données Actions, remplacez le `propertyName` par le nom du champ de votre schéma en cas de chameau. Par exemple, ACTION_EXPIRATION_TIMESTAMP serait et CREATION_TIMESTAMP serait CreationTimestamp. `actionExpirationTimestamp` Remplacez `property data` par les données de la propriété.

```
aws personalize-events put-actions \
  --dataset-arn dataset arn \
  --actions '[{
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    },
    {
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    }]'
```

Importer des actions individuellement (AWS SDKs)

Ajoutez une ou plusieurs actions à votre jeu de données Actions à l'aide de l' PutActions opération. Vous pouvez importer jusqu'à 10 actions en un seul `PutActions` appel. Si une action identique `actionId` figure déjà dans votre jeu de données Actions, Amazon Personalize la remplace par la nouvelle. Cette section part du principe que vous avez déjà créé un jeu de données Actions. Pour plus d'informations sur la création de jeux de données, consultezCréation d'un schéma et d'un jeu de données.

Le code suivant montre comment ajouter une ou plusieurs actions à votre jeu de données Actions. Pour chaque action, spécifiez le`actionId`. Si une action identique `actionId` figure déjà dans votre jeu de données Actions, Amazon Personalize la remplace par la nouvelle. En `properties` effet, pour chaque champ supplémentaire de votre jeu de données Actions, remplacez-le `propertyName` par le nom du champ indiqué dans votre schéma en cas de chameau. Par exemple, ACTION_EXPIRATION_TIMESTAMP serait et CREATION_TIMESTAMP serait CreationTimestamp. `actionExpirationTimestamp` Remplacez `property data` par les données de la propriété.

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_actions(
    datasetArn = 'dataset arn',
    actions = [{
```

```
      'actionId': 'actionId',
      'properties': "{\"propertyName\": \"property value\"}"
      },
      {
      'actionId': 'actionId',
      'properties': "{\"propertyName\": \"property value\"}"
      }]
)
```

Analyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize

Après avoir importé des données dans un ensemble de données d'interactions, d'utilisateurs ou d'articles, vous pouvez utiliser la console Amazon Personalize pour analyser les données. Vous pouvez en apprendre davantage sur vos données grâce à des informations sur les données et à des statistiques sur les colonnes et les lignes. Et vous pouvez découvrir les mesures que vous pouvez prendre pour améliorer vos données. Ces actions peuvent vous aider à répondre aux besoins en ressources d'Amazon Personalize, tels que les exigences en matière de formation des modèles, ou elles peuvent mener à de meilleures recommandations.

> **Important**
>
> Vous ne pouvez pas utiliser la console Amazon Personalize pour analyser les données d'une interaction Action ou d'un ensemble de données Actions.

Après avoir apporté les modifications recommandées, vous pouvez réimporter vos données et voir si vous avez résolu des problèmes ou amélioré les statistiques du jeu de données. Pour plus d'informations sur la mise à jour des données, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Si vous ne trouvez aucune information, vos données correspondent aux attentes d'Amazon Personalize en matière de données. Vous pouvez analyser les données d'un groupe de jeux de données de domaine ou d'un groupe de jeux de données personnalisé.

Lors de la génération d'informations et du calcul des statistiques, Amazon Personalize prend en compte toutes les données en masse et diffusées en continu provenant d'utilisateurs non anonymes. Les événements provenant d'utilisateurs anonymes ne sont pas pris en compte tant que vous ne les associez pas à un`userId`. Pour de plus amples informations, veuillez consulter Enregistrement d'événements pour les utilisateurs anonymes.

Rubriques

- Autorisations requises pour analyser les données
- Informations sur les données
- Afficher les informations et les statistiques des jeux de données

Autorisations requises pour analyser les données

Si vous accordez aux utilisateurs un accès complet à Amazon Personalize, aucune modification des autorisations n'est requise. Si vous accordez à vos utilisateurs uniquement les autorisations requises pour effectuer une tâche dans Amazon Personalize, votre politique Gestion des identités et des accès AWS (IAM) doit inclure les actions supplémentaires d'analyse des données suivantes.

- personnaliser : CreateDataInsightsJob
- personnaliser : ListDataInsightsJobs
- personnaliser : DescribeDataInsightsJob
- personnaliser : GetDataInsights

Informations sur les données

Vous trouverez ci-dessous les informations relatives aux données que vous pouvez générer dans Amazon Personalize.

Informations	Action	Ensemble de données associé (s)
Le jeu de données Interactions ne contient que X interactions. La formation des modèles nécessite un minimum de 1 000 interactions. Nous recommandons au moins 50 000.	Importez Y enregistrements d'interactions uniques supplémentaires avant d'entraîner un modèle.	Interactions entre les objets
Le jeu de données Interactions ne compte que X utilisateurs uniques ayant au moins deux interactions. La formation des modèles nécessite au moins 25 utilisateurs de ce type. Nous en recommandons au moins 1 000.	Importez au moins 2 enregistrements d'interactions chacun pour Y utilisateurs supplémentaires.	Interactions entre les objets

Informations	Action	Ensemble de données associé (s)
X % des éléments du jeu de données Items n'ont aucune interaction dans le jeu de données Interactions. Ils ne sont donc peut-être pas recommandés.	Assurez-vous d'importer toutes vos données d'interactions et de vérifier qu'il n'y a pas de différence IDs entre vos éléments et les ensembles de données d'interactions. Consultez les statistiques du jeu de données ci-dessous pour vos éléments et vos ensembles de données d'interactions afin de vous assurer que vous avez importé le nombre de lignes attendu. Si votre cas d'utilisation ou votre recette utilise l'exploration, modifiez la configuration d'exploration pour recommander d'autres éléments sans données d'interaction.	Interactions entre objets et objets
X % des utilisateurs du jeu de données Utilisateurs n'ont aucune interaction dans le jeu de données Interactions. Ces utilisateurs recevront des recommandations pour les articles les plus populaires.	Assurez-vous d'importer toutes vos données d'interactions et de vérifier qu'il n'y a pas de différence IDs entre vos utilisateurs et les ensembles de données d'interactions. Consultez les statistiques du jeu de données ci-dessous pour vos utilisateurs et vos ensembles de données d'interactions afin de vous assurer que vous avez importé le nombre de lignes attendu. Importez toutes les interactions supplémentaires afin qu'un plus grand nombre d'utilisateurs disposent de données sur les interactions.	Interactions avec les objets et utilisateurs

Informations	Action	Ensemble de données associé (s)
L'<Users or Items or Interactions>ensemble de données contient X % de lignes avec une valeur manquante. Cela peut avoir une incidence négative sur les recommandations. Nous recommandons que tous les champs obligatoires et facultatifs soient remplis à au moins 70 %.	Importez des enregistrements complets supplémentaires, ou réimportez des données sans lignes incomplètes, ou réimportez des données avec les valeurs manquantes remplacées par des données de remplacement, telles que la moyenne pour les colonnes numériques ou la valeur la plus courante pour les colonnes catégorielles.	N'importe lequel
Les colonnes suivantes de l'<datase tType>ensemble de données sont complétées à moins de 70 % : <ColumnName, ColumnName... >. Si ces données sont incluses dans la formation , elles peuvent avoir une incidence négative sur les recommandations. Nous recommandons que les colonnes qui autorisent les valeurs nulles soient remplies à au moins 70 %.	Importez des enregistrements complets supplémentaires, ou réimportez des données sans lignes incomplètes, ou réimportez des données avec les valeurs manquantes remplacées par des données de remplacement, telles que la moyenne pour les colonnes numériques ou la valeur la plus courante pour les colonnes catégorielles.	N'importe lequel
Les colonnes (numériques) suivantes présentent des valeurs aberrantes : <ColumnName, ColumnName... >. Les valeurs aberrantes ne constituent pas toujours un problème, mais elles ont parfois un impact négatif sur les recommandations.	À l'aide des statistiques des colonnes ci-dessous, vérifiez si les valeurs minimale et maximale de ces colonnes correspondent à vos attentes. Si ces valeurs sont inattendues, vérifiez que les données de ces colonnes ne sont pas inexactes et examinez votre collecte et votre traitement des données pour détecter tout problème.	N'importe lequel

Informations	Action	Ensemble de données associé (s)
Les colonnes suivantes comporten t plus de 1000 catégories possibles : <ColumnName, ColumnName... >. Si ces données sont incluses dans la formation, elles peuvent avoir un impact négatif sur les recommandations : <ColumnName, ColumnName... >.	Vérifiez vos données catégorielles pour détecter les problèmes, tels que les doublons de catégories dus à des variations orthographiques. Corrigez les inexactitudes et importez à nouveau les données.	N’importe lequel
Les colonnes de métadonnées textuelle s suivantes sont complétées à moins de 85 % et ne seront pas utilisées pour l'apprentissage des modèles : <ColumnName, ColumnName... >.	Importez des lignes supplémentaires ou importez à nouveau les lignes avec les données de texte pour ces colonnes.	Éléments
Le jeu de données Interactions contient plus de 10 types d'événements uniques, ce qui entraînera l'échec de l'entraîn ement du modèle.	Vérifiez que votre colonne de type d'événement ne contient pas d'inexact itudes, telles que des types d'événements dupliqués dus à des variations orthograp hiques. Supprimez les types d'événeme nts inutiles et réimportez les données.	Interacti ons entre les objets
Le jeu de données Interactions possède le même horodatage pour tous les enregistrements. Si vous utilisez une recette USER_SEGMENTATION et que tous les enregistrements ont le même horodatage, l'apprentissage du modèle échouera.	Vérifiez que vos données ne présenten t aucun problème d'horodatage et remplacez les horodatages dupliqués par des horodatages uniques.	Interacti ons entre les objets

Afficher les informations et les statistiques des jeux de données

Pour consulter des informations et des statistiques sur vos données dans les ensembles de données Amazon Personalize, accédez à vos ensembles de données dans la console Amazon Personalize et choisissez Exécuter l'analyse.

> **Note**
>
> Vous devez d'abord importer les données avant qu'Amazon Personalize puisse les analyser. Pour de plus amples informations, consultez Importation de données de formation dans les ensembles de données Amazon Personalize.

Pour consulter les informations et les statistiques

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sous Ensembles de données, sélectionnez Analyse des données.
4. En haut à droite, sélectionnez Exécuter l'analyse. Amazon Personalize commence à analyser vos données. Ce processus peut prendre jusqu'à 15 minutes. En cas de succès, les résultats apparaissent sur cette page.
5. Dans Insights, utilisez ce qui suit pour filtrer les informations qui apparaissent.
   - Pour trouver des informations qui incluent une langue spécifique, entrez vos critères dans Find Insight. Au fur et à mesure que vous saisissez du texte, la liste est mise à jour pour inclure uniquement les informations dont la chaîne est exacte dans l'aperçu ou l'action recommandée.
   - Pour filtrer les informations par type de jeu de données, remplacez Tous les ensembles de données par le type de jeu de données spécifique. La liste est mise à jour pour inclure uniquement les informations relatives à cet ensemble de données.
6. Pour consulter les statistiques d'un ensemble de données, procédez comme suit.
   - Pour afficher les informations générales et les statistiques relatives à un ensemble de données, telles que le nombre de lignes, d'utilisateurs uniques et d'éléments uniques dans un jeu de données Interactions, développez la section dédiée à l'ensemble de données.

- Pour afficher les statistiques détaillées d'une colonne, développez la section du jeu de données, choisissez Statistiques au niveau de la colonne et cliquez sur le bouton radio correspondant à la colonne.

7. Corrigez tout problème lié à vos données, réimportez-les et effectuez une autre analyse pour vérifier. Pour plus d'informations sur la réimportation de données, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Recommandeurs de domaines dans Amazon Personalize

Si vous avez créé un groupe de jeux de données de domaine, après avoir importé les données, vous êtes prêt à créer des recommandations pour les cas d'utilisation de votre domaine. Un recommandeur est une ressource de groupe de jeux de données de domaine qui génère des recommandations. Vous utilisez un outil de recommandation dans votre application pour obtenir des recommandations en temps réel concernant l'GetRecommendationsopération.

Lorsque vous créez un recommandeur, vous spécifiez un cas d'utilisation et Amazon Personalize forme les modèles qui soutiennent le recommandeur avec les meilleures configurations pour ce cas d'utilisation. Chaque cas d'utilisation comporte des exigences d'API différentes pour obtenir des recommandations. Pour obtenir une liste des cas d'utilisation des recommandations par domaine, consultezChoix d'un cas d'utilisation. Vous pouvez créer au maximum 15 recommandeurs par région.

Amazon Personalize reforme automatiquement les modèles qui soutiennent vos recommandations tous les 7 jours. Il s'agit d'un recyclage complet qui crée des modèles entièrement nouveaux basés sur l'intégralité des données de vos ensembles de données. Amazon Personalize met à jour les modèles existants toutes les deux heures pour inclure de nouveaux articles dans les recommandations avec exploration, en proposant des exemples d'utilisation « Top picks for you » et « Recommended for you ».

Lorsque vous créez un recommandeur, vous pouvez effectuer les opérations suivantes :

- Activez les métadonnées des éléments dans les recommandations. Pour de plus amples informations, veuillez consulter Activation des métadonnées dans les recommandations.
- Configurez les colonnes utilisées lors de l'entraînement des modèles qui soutiennent votre recommandateur. Pour de plus amples informations, veuillez consulter Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize.
- Pour les meilleurs choix pour vos cas d'utilisation ou ceux qui sont recommandés pour vous, vous pouvez configurer l'exploration. Pour de plus amples informations, veuillez consulter Configuration de l'exploration pour un recommandeur de domaine.

Après avoir créé un outil de recommandation, vous pouvez effectuer les opérations suivantes :

- Évaluez le recommandeur : vous pouvez évaluer les performances de votre recommandeur à l'aide de mesures hors ligne et en ligne. Pour de plus amples informations, veuillez consulter Évaluation d'un outil de recommandation de domaine Amazon Personalize.

- Arrêter et redémarrer le recommandeur : si vous souhaitez suspendre la facturation d'un recommandeur actif, vous pouvez arrêter le recommandeur et le redémarrer ultérieurement. Pour de plus amples informations, veuillez consulter Arrêter un recommandeur.
- Mettre à jour la configuration du recommandeur : vous pouvez mettre à jour les colonnes que le recommandant utilise pendant la formation et mettre à jour la capacité de demande du recommandant. Pour de plus amples informations, veuillez consulter Mettre à jour un recommandeur.
- Supprimer le recommandateur : vous pouvez supprimer des recommandateurs lors de cette opération. DeleteRecommender Vous pouvez également supprimer un recommandeur depuis la page de détails du recommandant dans la console Amazon Personalize.

Rubriques

- Statuts des recommandeurs
- Nombre minimal de demandes de recommandation par seconde et auto-scaling
- Création de recommandeurs de domaines dans Amazon Personalize
- Activation des métadonnées dans les recommandations pour un recommandeur de domaine dans Amazon Personalize
- Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize
- Configuration de l'exploration pour un recommandeur de domaine
- Évaluation d'un outil de recommandation de domaine Amazon Personalize
- Mettre à jour un recommandeur
- Arrêter un recommandeur

Statuts des recommandeurs

Un recommandeur peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Pour obtenir le statut du recommandeur, accédez à la page des recommandations dans la console Amazon Personalize ou utilisez l'DescribeRecommenderopération.

Nombre minimal de demandes de recommandation par seconde et auto-scaling

> **Important**
>
> Un montant élevé `minRecommendationRequestsPerSecond` fera augmenter votre facture. Nous vous recommandons de commencer par 1 pour `minRecommendationRequestsPerSecond` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minRecommendationRequestsPerSecond` si nécessaire.

Lorsque vous créez un recommandeur, vous pouvez configurer le nombre minimal de demandes de recommandation par seconde. Le nombre minimal de demandes de recommandation par seconde (`minRecommendationRequestsPerSecond`) indique le débit de base de demandes de recommandation fourni par Amazon Personalize. La valeur par défaut minRecommendationRequests PerSecond est1. Une demande de recommandation est une `GetRecommendations` opération unique. Le débit de demandes est mesuré en demandes par seconde et Amazon Personalize utilise vos demandes par seconde pour calculer le nombre de demandes par heure et le prix d'utilisation de votre outil de recommandation.

Si le nombre de demandes par seconde augmente au-delà`minRecommendationRequestsPerSecond`, Amazon Personalize redimensionne automatiquement la capacité allouée à la hausse ou à la baisse, mais jamais en dessous. `minRecommendationRequestsPerSecond` Il y a un court délai lors de l'augmentation de la capacité, ce qui peut entraîner une perte de demandes.

Votre facture correspond au montant le plus élevé entre le nombre minimum de demandes par heure (basé sur minRecommendationRequestsPerSecond) ou le nombre réel de demandes. Le débit de demandes réel utilisé est calculé comme la moyenne sur une requests/second période d'une heure. Nous vous recommandons de commencer par la valeur par défaut`minRecommendationRequestsPerSecond`, de suivre votre utilisation à l'aide CloudWatch des métriques Amazon, puis de l'`minRecommendationRequestsPerSecond`augmenter si nécessaire.

Création de recommandeurs de domaines dans Amazon Personalize

Vous pouvez créer des recommandations à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Vous trouverez ci-dessous des étapes détaillées pour créer des recommandations à l'aide de la console Amazon Personalize et des exemples de code qui montrent comment créer un recommandeur avec uniquement les champs obligatoires.

- Pour des exemples de code montrant comment activer les métadonnées dans les recommandations, consultezActivation des métadonnées dans les recommandations.
- Pour des exemples de code montrant comment configurer les colonnes utilisées lors de l'entraînement des modèles supportant votre recommandateur, consultezConfiguration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize.
- Pour des exemples de code montrant comment configurer l'exploration pour les cas d'`Recommended for you`utilisation `Top picks for your` ou les cas d'utilisation, consultezConfiguration de l'exploration pour un recommandeur de domaine.

Rubriques

- Création de recommandeurs (console)
- Création d'un recommandeur ()AWS CLI
- Création d'un recommandeur ()AWS SDKs

Création de recommandeurs (console)

> **Important**
>
> Un montant élevé `minRecommendationRequestsPerSecond` fera augmenter votre facture. Nous vous recommandons de commencer par 1 pour `minRecommendationRequestsPerSecond` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minRecommendationRequestsPerSecond` si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de demandes de recommandation par seconde et auto-scaling.

Créez des recommandations pour chacun de vos cas d'utilisation avec la console Amazon Personalize comme suit. Si vous venez de créer votre groupe de jeux de données de domaine et que vous êtes déjà sur la page Vue d'ensemble, passez à l'étape 3.

Pour créer des recommandateurs

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données de domaine.
3. À l'étape 3, choisissez Utiliser des recommandations, <domain name>puis Créer des recommandations.
4. Sur la page Choisir des cas d'utilisation, choisissez les cas d'utilisation pour lesquels vous souhaitez créer des recommandations et attribuez à chacun un nom de recommandeur. Amazon Personalize crée un outil de recommandation pour chaque cas d'utilisation que vous choisissez. Les cas d'utilisation disponibles dépendent de votre domaine. Pour plus d'informations sur le choix d'un cas d'utilisation, voirChoix d'un cas d'utilisation.
5. Choisissez Suivant.
6. Sur la page Configuration avancée, configurez chaque recommandateur en fonction des besoins de votre entreprise :

   - Pour chaque ensemble de données utilisé par le cas d'utilisation du recommandeur, vous pouvez choisir les colonnes prises en compte par Amazon Personalize lors de la formation des modèles qui soutiennent votre recommandeur. Par défaut, Amazon Personalize utilise toutes les colonnes qui peuvent être utilisées lors de la formation. Pour de plus amples informations, veuillez consulter Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize.
   - Vous pouvez modifier le nombre minimum de demandes de recommandation par seconde afin de spécifier une nouvelle capacité minimale de demande pour votre recommandeur. Un montant élevé `minRecommendationRequestsPerSecond` fera augmenter votre facture. Nous vous recommandons de commencer par 1 (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minRecommendationRequestsPerSecond` si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de demandes de recommandation par seconde et auto-scaling.

- Si vous souhaitez pouvoir inclure les métadonnées du jeu de données Items dans les recommandations, choisissez Retourner les métadonnées des items dans les résultats des recommandations. Si cette option est activée, vous pouvez spécifier les colonnes de votre jeu de données Items dans votre demande de recommandations ou de classement personnalisé. Amazon Personalize renvoie ces données pour chaque article dans la réponse à la recommandation.

  Pour activer les métadonnées, vous devez disposer d'un jeu de données Items avec une colonne de métadonnées.

- Pour `Top picks for you` nos cas d'`Recommended for you`utilisation, modifiez éventuellement la configuration d'exploration. L'exploration consiste à tester différentes recommandations d'articles pour savoir comment les utilisateurs réagissent aux éléments avec très peu de données d'interaction. Utilisez les champs suivants pour configurer l'exploration :

  - Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
  - Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

    Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

- Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.

7. Pour créer des recommandations pour chacun de vos cas d'utilisation, choisissez Créer des recommandations.

   Vous pouvez suivre le statut de chaque recommandeur sur la page Recommandeurs. Lorsque le statut de votre recommandeur est Actif, vous pouvez l'utiliser dans votre application pour obtenir des recommandations.

Création d'un recommandeur ()AWS CLI

Utilisez le AWS CLI code suivant pour créer une recommandation pour un cas d'utilisation de domaine. Exécutez ce code pour chacun des cas d'utilisation de votre domaine. Pour`recipeArn`, fournissez l'Amazon Resource Name (ARN) correspondant à votre cas d'utilisation. Les cas d'utilisation disponibles dépendent de votre domaine. Pour une liste des cas d'utilisation et leur description, ARNs voirChoix d'un cas d'utilisation.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN
```

Création d'un recommandeur ()AWS SDKs

Créez un outil de recommandation pour un cas d'utilisation de domaine à l'aide du code suivant. Donnez un nom à votre recommandateur et indiquez le nom de ressource Amazon (ARN) de votre groupe de données de domaine. Pour`recipeArn`, fournissez l'ARN correspondant à votre cas d'utilisation. Exécutez ce code pour chacun des cas d'utilisation de votre domaine. Les cas d'utilisation disponibles dépendent de votre domaine. Pour obtenir une liste des cas d'utilisation, de leurs ARNs exigences et de leurs exigences, consultezChoix d'un cas d'utilisation.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
  name = 'recommender name',
  recipeArn = 'recipe ARN',
  datasetGroupArn = 'dataset group ARN'
```

```
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for Java 2.x

```
    public static String createRecommender(PersonalizeClient personalizeClient,
            String name,
            String datasetGroupArn,
            String recipeArn) {

        long maxTime = 0;
        long waitInMilliseconds = 30 * 1000; // 30 seconds
        String recommenderStatus = "";

        try {
            CreateRecommenderRequest createRecommenderRequest =
CreateRecommenderRequest.builder()
                    .datasetGroupArn(datasetGroupArn)
                    .name(name)
                    .recipeArn(recipeArn)
                    .build();

            CreateRecommenderResponse recommenderResponse = personalizeClient
                    .createRecommender(createRecommenderRequest);
            String recommenderArn = recommenderResponse.recommenderArn();
            System.out.println("The recommender ARN is " + recommenderArn);

            DescribeRecommenderRequest describeRecommenderRequest =
DescribeRecommenderRequest.builder()
                    .recommenderArn(recommenderArn)
                    .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                        .status();
                System.out.println("Recommender status: " + recommenderStatus);
```

```
                if (recommenderStatus.equals("ACTIVE") ||
 recommenderStatus.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return recommenderArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",                    /* required */
  recipeArn: "RECIPE_ARN",                     /* required */
  datasetGroupArn: "DATASET_GROUP_ARN"         /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
```

```
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Activation des métadonnées dans les recommandations pour un recommandeur de domaine dans Amazon Personalize

> **Important**
>
> Lorsque vous activez les métadonnées dans les recommandations, vous encourez des coûts supplémentaires. Pour plus d'informations, consultez les tarifs d'Amazon Personalize.

Lorsque vous créez un outil de recommandation, vous pouvez activer l'option permettant d'inclure les métadonnées des éléments de votre jeu de données d'articles dans les résultats des recommandations. Si cette option est activée, vous pouvez spécifier les colonnes de votre jeu de données Items dans votre demande de recommandations. Amazon Personalize renvoie ces données pour chaque article dans la réponse à la recommandation.

Vous pouvez utiliser les métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant les genres de films aux carrousels. Vous pouvez également l'utiliser pour évaluer visuellement la qualité des recommandations. Si vous utilisez l'IA générative dans votre application, vous pouvez intégrer les métadonnées aux instructions de l'IA pour générer un contenu plus pertinent. Pour plus d'informations sur l'utilisation d'Amazon Personalize avec l'IA générative, consultezAmazon Personalize et l'IA générative.

Pour ajouter des métadonnées aux recommandations, vous devez disposer d'un jeu de données Items avec une colonne de métadonnées. Vous n'êtes pas obligé d'utiliser les métadonnées lors de l'entraînement. Pour plus d'informations sur la création d'un jeu de données, consultezCréation d'un schéma et d'un jeu de données. Pour plus d'informations sur la mise à jour des données d'un ensemble de données, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Les exemples de code suivants montrent comment activer l'option permettant d'inclure les métadonnées des éléments avec le AWS CLI ou le AWS SDKs. Pour ce faire, avec la console

Amazon Personalize, vous activez les métadonnées sur la page de configuration avancée lorsque vous créez le recommandeur. Pour de plus amples informations, veuillez consulter Création de recommandeurs (console).

Activation des métadonnées (AWS CLI)

Si vous possédez un jeu de données Items et que vous souhaitez avoir l'option d'inclure des métadonnées lorsque vous recevez des recommandations, définissez `enableMetadataWithRecommendations` ce paramètre sur `true` dans le`recommender-config`.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group \
--recipe-arn recipe ARN \
--recommender-config "{\"enableMetadataWithRecommendations\": "true"}"
```

Activation des métadonnées (AWS SDK)

Si vous possédez un jeu de données Items et que vous souhaitez avoir l'option d'inclure des métadonnées lorsque vous recevez des recommandations, définissez `enableMetadataWithRecommendations` ce paramètre sur `true` dans le`recommender-config`.

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
  name = 'recommender name',
  recipeArn = 'recipe name',
  datasetGroupArn = 'dataset group ARN',
  recommenderConfig = {"enableMetadataWithRecommendations": True}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize

Lorsque vous créez un recommandeur, vous pouvez modifier les colonnes prises en compte par Amazon Personalize lors de la formation des modèles qui soutiennent votre recommandateur.

Vous pouvez le faire pour expérimenter différentes combinaisons de données d'entraînement. Vous pouvez également exclure des colonnes dépourvues de données significatives. Par exemple, vous pouvez avoir une colonne que vous souhaitez utiliser uniquement pour filtrer les recommandations. Vous pouvez exclure cette colonne de la formation et Amazon Personalize ne la prend en compte que lors du filtrage.

Vous ne pouvez pas exclure les colonnes EVENT_TYPE. Par défaut, Amazon Personalize utilise toutes les colonnes qui peuvent être utilisées lors de la formation. Les données suivantes sont toujours exclues de la formation :

- Colonnes avec le type de données booléen
- Données sur les impressions
- Champs de chaîne personnalisés qui ne sont ni catégoriques ni textuels

Vous ne pouvez pas inclure de données sur les impressions dans la formation, mais si votre cas d'utilisation ou votre recette les utilise, Amazon Personalize utilise les données d'impressions pour guider l'exploration lorsque vous recevez des recommandations.

Les exemples de code suivants montrent comment configurer les colonnes utilisées lors de l'entraînement avec le AWS CLI ou le AWS SDKs. Pour ce faire, avec la console Amazon Personalize, vous devez spécifier les colonnes à utiliser sur la page de configuration avancée lorsque vous créez le recommandateur. Pour de plus amples informations, veuillez consulter Création de recommandeurs (console).

Configuration des colonnes utilisées lors de l'entraînement (AWS CLI)

Pour exclure des colonnes de l'entraînement, fournissez l'`excludedDatasetColumns`objet dans le `trainingDataConfig` cadre de la configuration du recommandeur. Pour chaque clé de l'objet, indiquez le type de jeu de données. Pour chaque valeur, fournissez la liste des colonnes à exclure. Pour de plus amples informations, veuillez consulter Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

Configuration des colonnes utilisées lors de l'entraînement (AWS SDKs)

Pour exclure des colonnes de l'entraînement, fournissez l'`excludedDatasetColumns`objet dans le `trainingDataConfig` cadre de la configuration du recommandeur. Pour chaque clé, indiquez le type de jeu de données. Pour chaque valeur, fournissez la liste des colonnes à exclure. Le code suivant montre comment exclure des colonnes de l'entraînement lorsque vous créez un recommandateur. Pour de plus amples informations, veuillez consulter Configuration des colonnes utilisées lors de la création d'un système de recommandation de domaine Amazon Personalize.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
  name = 'recommender name',
  recipeArn = 'recipe name',
  datasetGroupArn = 'dataset group ARN',
  recommenderConfig = {
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "datasetType": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",             /* required */
  recipeArn: "RECIPE_ARN",              /* required */
  datasetGroupArn: "DATASET_GROUP_ARN", /* required */
  recommenderConfig: {
    trainingDataConfig: {
      excludedDatasetColumns: {
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Configuration de l'exploration pour un recommandeur de domaine

Pour nos cas `Top picks for your` d'`Recommended for you`utilisation, Amazon Personalize utilise l'exploration lorsqu'il recommande des articles. L'exploration consiste à tester différentes

recommandations d'articles pour savoir comment les utilisateurs réagissent aux éléments avec très peu de données d'interaction. Vous pouvez configurer l'exploration comme suit :

- Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
- Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

  Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

Les exemples de code suivants montrent comment configurer l'exploration pour un recommandeur avec le AWS CLI ou le AWS SDKs. Pour ce faire avec la console Amazon Personalize, vous devez spécifier la configuration d'exploration sur la page de configuration avancée lorsque vous créez le recommandeur. Pour de plus amples informations, veuillez consulter Création de recommandeurs (console).

Configuration de l'exploration (AWS CLI)

Le code suivant montre comment configurer l'exploration lorsque vous créez une recommandation pour le cas `Top picks for you` d'utilisation. L'exemple utilise les valeurs par défaut.

Si vous possédez un jeu de données Items et que vous souhaitez avoir la possibilité d'inclure des métadonnées lorsque vous recevez des recommandations, mettez-le `recommender-config` à jour pour ajouter un `enableMetadataWithRecommendations` champ et définissez-le sur`true`.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn arn:aws:personalize:::recipe/aws-vod-top-picks \
--recommender-config "{\"itemExplorationConfig\":{\"explorationWeight\":\"0.3\",
\"explorationItemAgeCutOff\":\"30\"}}"
```

Configuration de l'exploration (AWS SDKs)

Pour nos cas `Top picks for your` d'`Recommended for you`utilisation, Amazon Personalize utilise l'exploration lorsqu'il recommande des articles. L'exploration consiste à tester différentes recommandations d'articles pour savoir comment les utilisateurs réagissent aux éléments avec très peu de données d'interaction. Vous pouvez configurer l'exploration comme suit :

- Accent mis sur l'exploration d'éléments moins pertinents (poids d'exploration) — Configurez la quantité à explorer. Spécifiez une valeur décimale comprise entre 0 et 1. La valeur par défaut est 0,3. Plus la valeur est proche de 1, plus l'exploration est intense. Avec plus d'exploration, les recommandations incluent davantage d'éléments avec moins de données d'interactions entre les éléments ou de pertinence en fonction du comportement antérieur. À zéro, aucune exploration n'a lieu et les recommandations sont basées sur les données actuelles (pertinence).
- Âge limite des objets d'exploration : spécifiez l'âge maximum des objets en jours depuis la dernière interaction entre tous les objets du jeu de données des interactions entre les objets. Cela définit la portée de l'exploration des objets en fonction de leur âge. Amazon Personalize détermine l'âge de l'article en fonction de son horodatage de création ou, si les données d'horodatage de création sont manquantes, des données relatives aux interactions entre les articles. Pour plus d'informations sur la façon dont Amazon Personalize détermine l'âge des articles, consultezDonnées d'horodatage de création.

  Pour augmenter le nombre d'articles pris en compte par Amazon Personalize lors de l'exploration, entrez une valeur supérieure. Le minimum est de 1 jour et la valeur par défaut est de 30 jours. Les recommandations peuvent inclure des articles dont l'âge est supérieur à l'âge limite que vous spécifiez. Cela est dû au fait que ces éléments sont pertinents pour l'utilisateur et que l'exploration ne les a pas identifiés.

Le code suivant montre comment configurer l'exploration lorsque vous créez un recommandateur. L'exemple utilise les valeurs par défaut.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
  name = 'recommender name',
  recipeArn = 'arn:aws:personalize:::recipe/aws-vod-top-picks',
  datasetGroupArn = 'dataset group ARN',
  recommenderConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
 "explorationItemAgeCutOff": "30"}}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",                    /* required */
  recipeArn: "RECIPE_ARN",                     /* required */
  datasetGroupArn: "DATASET_GROUP_ARN",        /* required */
  recommenderConfig: {
    itemExplorationConfig: {
      explorationWeight: "0.3",
      explorationItemAgeCutOff: "30"
    }
  }
};

export const run = async () => {
```

```
  try {
    const response = await personalizeClient.send(new
 CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Évaluation d'un outil de recommandation de domaine Amazon Personalize

Vous pouvez évaluer les performances de votre recommandeur à l'aide de mesures hors ligne et en ligne. Les métriques en ligne sont les résultats empiriques que vous observez dans les interactions de vos utilisateurs avec des recommandations en temps réel. Par exemple, vous pouvez enregistrer le taux de clics de vos utilisateurs lorsqu'ils parcourent votre catalogue. Vous êtes responsable de la génération et de l'enregistrement de tous les indicateurs en ligne.

Les métriques hors ligne sont celles qu'Amazon Personalize génère lorsque vous créez un recommandeur. Grâce aux métriques hors ligne, vous pouvez évaluer les performances des modèles qui soutiennent votre recommandation. Vous pouvez visualiser les effets de la modification de la configuration d'un recommandeur, et vous pouvez comparer les résultats obtenus par des recommandeurs formés à différents cas d'utilisation avec les mêmes données dans le même groupe de données.

Évitez de comparer les indicateurs de différents recommandeurs formés avec des données différentes. La différence entre les métriques peut être due à la différence entre les données plutôt qu'aux performances du modèle. Par exemple, vous pouvez avoir un groupe de jeux de données contenant des données d'`purchase`événements éparses pour chaque utilisateur, et un autre contenant des données d'`view`événements robustes. Sur la base de mesures telles que `precision at K` le recommandateur formé à la visualisation des données d'événements peut sembler à tort plus performant en raison du nombre plus élevé d'interactions.

Pour obtenir des indicateurs de performance, Amazon Personalize divise les données d'interaction en entrée en un ensemble d'entraînement et un ensemble de tests. Le kit de formation comprend 90 %

des données de vos utilisateurs et de leurs interactions. L'ensemble de tests comprend les 10 % d'utilisateurs restants et leurs données d'interactions.

Amazon Personalize crée ensuite le recommandeur à l'aide du kit d'entraînement. Une fois la formation terminée, Amazon Personalize fournit au nouveau recommandant les 90 % les plus anciens des données de chaque utilisateur issues du set de test en entrée. Amazon Personalize calcule ensuite les indicateurs en comparant les recommandations générées par le recommandeur aux interactions réelles figurant dans les 10 % les plus récents des données de chaque utilisateur issues du kit de test.

Rubriques

- Récupération des métriques
- Définitions des métriques
- Exemple
- Ressources supplémentaires

Récupération des métriques

Une fois que votre recommandeur est actif, vous pouvez consulter ses indicateurs dans la console Amazon Personalize ou récupérer des indicateurs en appelant l'DescribeRecommenderopération.

Rubriques

- Affichage des métriques (console)
- Récupération de métriques (AWS CLI)
- Récupération de métriques (AWS SDK)

Affichage des métriques (console)

Pour consulter les statistiques du recommandant dans la console, vous accédez à la page de détails de votre recommandeur.

1. Ouvrez la console Amazon Personalize à l'adresse https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données de domaine.

3. Dans le volet de navigation, sélectionnez Recommenders.
4. Dans la liste des recommandateurs, choisissez celui dont vous souhaitez consulter les statistiques.

Récupération de métriques (AWS CLI)

Le code suivant montre comment obtenir des métriques pour un recommandeur à l'aide du AWS CLI.

```
aws personalize describe-recommender \
--recommender-arn recommender arn
```

Voici un exemple des mesures produites par un outil de recommandation créé pour le cas d'utilisation Top picks for you pour le domaine VIDEO_ON_DEMAND.

```
{
    "recommender": {
        "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
        "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
        "name": "name123",
        "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
        "modelMetrics": {
            "coverage": 0.27,
            "mean_reciprocal_rank_at_25": 0.0379,
            "normalized_discounted_cumulative_gain_at_5": 0.0405,
            "normalized_discounted_cumulative_gain_at_10": 0.0513,
            "normalized_discounted_cumulative_gain_at_25": 0.0828,
            "precision_at_5": 0.0136,
            "precision_at_10": 0.0102,
            "precision_at_25": 0.0091,
        }
        "recommenderConfig": {},
        "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
        "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
        "status": "ACTIVE",
    }
}
```

Récupération de métriques (AWS SDK)

Le code suivant montre comment obtenir des métriques pour un recommandeur à l'aide du AWS SDK pour Python (Boto3).

```
import boto3

personalize = boto3.client('personalize')

response = personalize.describe_recommender(
    recommenderArn = 'recommender_arn'
)
print(response['recommender']['modelMetrics'])
```

Voici un exemple des métriques produites par un outil de recommandation créé pour le cas d'utilisation Top picks for you pour le domaine VIDEO_ON_DEMAND.

```
{
    "recommender": {
        "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
        "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
        "name": "name123",
        "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
        "modelMetrics": {
            "coverage": 0.27,
            "mean_reciprocal_rank_at_25": 0.0379,
            "normalized_discounted_cumulative_gain_at_5": 0.0405,
            "normalized_discounted_cumulative_gain_at_10": 0.0513,
            "normalized_discounted_cumulative_gain_at_25": 0.0828,
            "precision_at_5": 0.0136,
            "precision_at_10": 0.0102,
            "precision_at_25": 0.0091,
        }
        "recommenderConfig": {},
        "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
        "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
        "status": "ACTIVE",
    }
}
```

Définitions des métriques

Les statistiques générées par Amazon Personalize pour les recommandateurs sont décrites ci-dessous en utilisant les termes suivants :

- Une recommandation pertinente est une recommandation pour un élément avec lequel l'utilisateur a réellement interagi. Ces éléments proviennent des 10 % les plus récents des données d'interactions de chaque utilisateur issues du kit de test.
- Rang fait référence à la position d'un élément dans la liste des recommandations. La position 1 (en haut de la liste) est présumée comme étant la plus pertinente pour l'utilisateur.

Pour chaque métrique, les valeurs les plus élevées (plus proches de 1) sont préférables. Pour en savoir plus, consultez les ressources répertoriées dansRessources supplémentaires.

couverture

La valeur de la couverture indique la proportion d'articles uniques qu'Amazon Personalize peut recommander par rapport au nombre total d'articles uniques figurant dans les ensembles de données Interactions et Articles. Un score de couverture plus élevé signifie qu'Amazon Personalize recommande un plus grand nombre de vos articles, plutôt que les mêmes articles à plusieurs reprises pour différents utilisateurs. Les cas d'utilisation qui proposent l'exploration d'objets, tels que les meilleurs choix pour vous (VIDEO_ON_DEMAND) et les articles recommandés pour vous (ECOMMERCE), sont mieux couverts que ceux qui ne le sont pas.

rang réciproque moyen à 25

Cette métrique vous renseigne sur la capacité d'un modèle à générer une recommandation pertinente à la première place. Vous pouvez choisir un modèle dont le classement réciproque moyen est élevé (25) si vous générez des résultats de recherche pertinents pour un utilisateur et ne vous attendez pas à ce que l'utilisateur choisisse un élément plus bas dans la liste. Par exemple, les utilisateurs choisissent fréquemment la première recette de cuisine dans les résultats de recherche.

Amazon Personalize calcule cette métrique en utilisant le score de classement réciproque moyen pour les demandes de recommandations. Chaque score de classement réciproque est calculé comme suit :`1 / the rank of the highest item interacted with by the user`, où le total des classements possibles est de 25. Les autres éléments de rang inférieur avec lesquels l'utilisateur interagit sont ignorés. Si l'utilisateur a choisi le premier élément, le score est de 1. S'ils ne choisissent aucun objet, le score est de 0.

Par exemple, vous pouvez présenter 25 recommandations à trois utilisateurs différents :

- Si l'utilisateur 1 clique sur l'élément au rang 4 et sur l'élément au rang 10, son score de classement réciproque est 1/4.
- Si l'utilisateur 2 clique sur un élément au rang 2, un élément au rang 4 et un élément au rang 12, son score de classement réciproque est 1/2.
- Si l'utilisateur 3 clique sur un seul élément au rang 6, son score de classement réciproque est 1/6.

Le classement réciproque moyen de toutes les demandes de recommandations (dans ce cas 3) est calculé comme suit`(1/4 + 1/2 + 1/6) / 3 = .3056`.

gain cumulé actualisé normalisé (NDCG) à K (5, 10 ou 25)

Cette métrique indique dans quelle mesure votre modèle classe les recommandations, K étant un échantillon de 5, 10 ou 25 recommandations. Cette métrique est utile si vous êtes particulièrement intéressé par le classement des recommandations au-delà de l'élément le mieux classé (pour cela, voir`mean reciprocal rank at 25`). Par exemple, le score de `NDCG at 10` serait utile si vous avez une application qui affiche jusqu'à 10 films dans un carrousel à la fois.

Amazon Personalize calcule le NDCG en attribuant une pondération aux recommandations en fonction de leur position dans le classement de chaque utilisateur du kit de test. Chaque recommandation est réduite (avec une pondération plus faible) selon un facteur qui dépend de sa position. La métrique finale est la moyenne de tous les utilisateurs du kit de test. Le gain cumulé actualisé normalisé à K suppose que les recommandations situées en bas d'une liste sont moins pertinentes que les recommandations situées en haut de la liste.

Amazon Personalize utilise un facteur de pondération de`1/log(1 + position)`, où le haut de la liste est la position`1`.

précision à K

Cette métrique indique dans quelle mesure les recommandations de votre modèle sont pertinentes sur la base d'un échantillon de K (5, 10 ou 25) recommandations.

Amazon Personalize calcule cette métrique en fonction du nombre de recommandations pertinentes parmi les K meilleures recommandations pour chaque utilisateur du kit de test, divisé par K, où K est 5, 10 ou 25. La mesure finale est la moyenne de tous les utilisateurs du kit de test.

Par exemple, si vous recommandez 10 éléments à un utilisateur et que l'utilisateur interagit avec 3 d'entre eux, la précision à K est de 3 éléments correctement prédits divisés par le total des 10 éléments recommandés :`3 / 10 = .30`.

Cette métrique récompense la recommandation précise d'articles pertinents. Plus le score est proche de 1, plus le modèle est précis.

Exemple

Voici un exemple simple de recommandation qui produit une liste de recommandations pour un utilisateur spécifique. Les deuxième et cinquième recommandations correspondent aux enregistrements des données de test pour cet utilisateur. Voici les recommandations pertinentes. Si `K` est défini sur 5, les métriques suivantes sont générées pour l'utilisateur.

reciprocal_rank

Calcul : 1/2

Résultat : 0,5000

normalized_discounted_cumulative_gain_at_5

Calcul : 1/log ((1 + 2) + 1/log (1 + 5))/(1/log(1 + 1) + 1/log (1 + 2))

Résultat : 0,6241

precision_at_5

Calcul : 2/5

Résultat : 0,4000

Ressources supplémentaires

Pour en savoir plus sur les différents types de métriques pour les systèmes de recommandation, consultez les ressources externes suivantes :

- MRR vs MAP vs NDCG : métriques Rank-Aware d'évaluation et quand les utiliser
- Gain cumulé réduit : les indicateurs de classement que vous devez connaître
- Rappel et précision à k pour les systèmes de recommandation

- Mesures d'évaluation du classement pour les systèmes de recommandation

Mettre à jour un recommandeur

Après avoir créé un recommandeur, vous pouvez mettre à jour sa configuration :

- Vous pouvez mettre à jour les colonnes utilisées par le recommandateur lors de l'entraînement. Si vous modifiez les colonnes utilisées lors de la formation, Amazon Personalize lance automatiquement un réentraînement complet des modèles qui soutiennent votre conseiller. Tant que la mise à jour est terminée, vous pouvez toujours obtenir des recommandations de la part du recommandeur. Le recommandeur utilise la configuration précédente jusqu'à ce que la mise à jour soit terminée. Pour suivre l'état de cette mise à jour, utilisez le résultat `latestRecommenderUpdate` renvoyé lors de l'DescribeRecommenderopération. Si vous fournissez les mêmes colonnes que celles que vous avez fournies lors de la création du recommandateur, aucune mise à jour n'est effectuée.
- Vous pouvez mettre à jour le nombre minimal de demandes de recommandation par seconde du recommandeur. Cela indique le débit de base des demandes de recommandation fourni par Amazon Personalize. Une valeur élevée augmentera votre facture. Nous vous recommandons de commencer par 1. Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-la si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de demandes de recommandation par seconde et auto-scaling.
- Pour les cas d'utilisation les plus intéressants pour vous et les cas d'utilisation recommandés pour vous, vous pouvez mettre à jour la configuration d'exploration en mettant l'accent sur l'exploration des objets pertinents et en ajustant l'âge limite des objets d'exploration. Pour plus d'informations sur l'exploration, consultez la section correspondant à votre cas d'utilisation dansChoix d'un cas d'utilisation.

Vous pouvez mettre à jour les recommandations à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Rubriques

- Mettre à jour un recommandeur (console Amazon Personalize)
- Mettre à jour un recommandeur ()AWS CLI
- Mettre à jour un recommandeur ()AWS SDKs

Mettre à jour un recommandeur (console Amazon Personalize)

Après avoir créé un recommandeur, vous pouvez le mettre à jour. Vous pouvez mettre à jour les colonnes que le recommandeur utilise pendant l'entraînement et le nombre minimal de demandes de recommandation par seconde du recommandant. Pour les cas d'utilisation les plus intéressants pour vous et les cas d'utilisation recommandés pour vous, vous pouvez mettre à jour la configuration d'exploration. Pour mettre à jour un outil de recommandation avec la console, procédez comme suit.

Pour mettre à jour la configuration d'un recommandeur (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données de domaine.
3. Dans le volet de navigation, sélectionnez Recommenders.
4. Sur la page Recommandeurs, choisissez le recommandeur que vous souhaitez mettre à jour.
5. Dans la configuration de Recommender, choisissez Modifier.
6. Modifiez la configuration du recommandeur et choisissez Mettre à jour. Pour plus d'informations sur les différentes options de configuration, consultezCréation de recommandeurs (console).

Mettre à jour un recommandeur ()AWS CLI

Pour mettre à jour le recommandeur avec le AWS CLI, utilisez la `update-recommender` commande. Fournissez le nom de ressource Amazon (ARN) pour le recommandateur et la configuration mise à jour. Le code suivant montre comment mettre à jour les colonnes qu'un recommandateur utilise pour la formation.

```
aws personalize update-recommender \
--dataset-group-arn dataset group ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

Si vous modifiez les colonnes utilisées lors de la formation, Amazon Personalize lance automatiquement un réentraînement complet des modèles qui soutiennent votre recommandateur. Tant que la mise à jour est terminée, vous pouvez toujours obtenir des recommandations de la part du recommandeur. Le recommandeur utilise la configuration précédente jusqu'à ce que la mise à jour

soit terminée. Pour suivre l'état de cette mise à jour, utilisez le résultat `latestRecommenderUpdate` renvoyé lors de l'DescribeRecommenderopération.

Pour plus d'informations sur les différentes configurations que vous pouvez modifier, consultezRecommenderConfig.

Mettre à jour un recommandeur ()AWS SDKs

Pour mettre à jour le recommandeur avec le AWS, utilisez l'UpdateRecommenderopération. Fournissez le nom de ressource Amazon (ARN) du recommandeur et spécifiez la nouvelle configuration. Le code suivant montre comment mettre à jour les colonnes qu'un recommandateur utilise pour la formation.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

update_recommender_response = personalize.update_recommender(
  recommenderArn = 'dataset group ARN',
  recommenderConfig = {
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "datasetType": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { UpdateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request's parameters
```

```
export const updateRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN", /* required */
  recommenderConfig: {
    trainingDataConfig: {
      excludedDatasetColumns: {
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 UpdateRecommenderCommand(updateRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Si vous modifiez les colonnes utilisées lors `excludedDatasetColumns` de la formation dans le`recommenderConfig`, Amazon Personalize lance automatiquement un réentraînement complet des modèles soutenant votre recommandateur. Tant que la mise à jour est terminée, vous pouvez toujours obtenir des recommandations de la part du recommandeur. Le recommandeur utilise la configuration précédente jusqu'à ce que la mise à jour soit terminée. Pour suivre l'état de cette mise à jour, utilisez le résultat `latestRecommenderUpdate` renvoyé lors de l'DescribeRecommenderopération.

Pour plus d'informations sur les différentes configurations que vous pouvez modifier, consultezRecommenderConfig.

Arrêter un recommandeur

Une fois que votre recommandeur est actif, vous pouvez l'arrêter et le démarrer plus tard. Ainsi, vous pouvez suspendre la facturation des recommandations et ne payer que lorsque vous l'utilisez. Par exemple, il se peut que vous deviez obtenir des recommandations uniquement certains jours de

la semaine. Vous pouvez arrêter le recommandeur les jours où vous n'en avez pas besoin, puis le démarrer les jours où vous en avez besoin.

Une fois que vous avez arrêté un outil de recommandation, vous ne pouvez pas l'utiliser pour obtenir des recommandations. L'arrêt d'un recommandant met fin à la facturation et à la reconversion du recommandant. Toutefois, l'arrêt d'un recommandant ne le supprime pas. Vous pouvez le redémarrer à tout moment et recommencer à recevoir des recommandations. Le lancement d'un recommandeur n'entraîne pas la création d'un nouveau recommandeur à partir de vos données. Il reprend plutôt la facturation et le recyclage des recommandeurs tous les 7 jours.

Vous pouvez arrêter et démarrer un programme de recommandation à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI), AWS SDKs.

États recommandant

Lorsque vous arrêtez une recommandation, son état passe d'ACTIF à INACTIF dans l'ordre suivant :

ACTIF > ARRÊT EN ATTENTE > ARRÊT EN COURS > INACTIF

Lorsque vous démarrez une recommandation, son état passe de INACTIF à ACTIF dans l'ordre suivant :

INACTIF > DÉBUT EN ATTENTE > DÉMARRAGE EN COURS > ACTIF

Rubriques

- Arrêter un recommandeur (console)
- Arrêter un recommandeur ()AWS CLI
- Arrêter un recommandeur ()AWS SDKs

Arrêter un recommandeur (console)

Vous pouvez arrêter un recommandeur depuis la page de détails du recommandant dans la console Amazon Personalize.

Pour arrêter un recommandant

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.

2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données de domaine.
3. Dans le volet de navigation, sélectionnez Recommenders.
4. Sur la page Recommandeurs, choisissez le recommandeur que vous souhaitez arrêter.
5. Sur la page de détails du recommandateur, choisissez Arrêter le recommandeur en haut à droite et confirmez dans la fenêtre qui s'affiche. Lorsque le statut de recommandation est inactif, cela signifie que votre recommandeur s'est arrêté. Vous pouvez le recommencer à partir de la même page.

Arrêter un recommandeur ()AWS CLI

Pour arrêter un recommandeur actif avec le AWS CLI, utilisez la `stop-recommender` commande, qui utilise l'opération d'StopRecommenderAPI, et fournissez l'Amazon Resource Name (ARN) pour le recommandateur. Pour le redémarrer, vous pouvez utiliser `start-recommender` la commande, qui utilise leStartRecommender. Le code suivant montre comment arrêter un recommandeur :

```
aws personalize stop-recommender --recommender-arn "recommender arn"
```

Arrêter un recommandeur ()AWS SDKs

Pour arrêter un recommandeur actif avec le AWS SDKs, utilisez l'opération StopRecommender API et fournissez l'Amazon Resource Name (ARN) pour le recommandeur. Pour le redémarrer, vous utilisez leStartRecommender. Le code suivant montre comment arrêter un recommandeur :

SDK for Python (Boto3)

Pour arrêter un recommandeur actif avec le SDK pour Python (Boto3), utilisez la méthode et fournissez `stop_recommender` l'Amazon Resource Name (ARN) pour le recommandateur comme suit.

```
import boto3
personalize = boto3.client('personalize')

stop_recommender_response = personalize.stop_recommender(
    recommenderArn = "recommenderARN"
)
print(stop_recommender_response)
```

SDK for Java 2.x

Pour arrêter un recommandeur actif avec le SDK pour Java 2.x, utilisez `stopRecommender` la méthode et fournissez l'ARN du recommandateur comme suit.

```
public static void stopRecommender(PersonalizeClient personalizeClient,
                                                String datasetGroupArn) {

    try {

        StopRecommenderRequest stopRecommenderRequest =
 StopRecommenderRequest.builder()
                .recommenderArn(recommenderArn)
                .build();
        personalizeClient.stopRecommender(stopRecommenderRequest);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { StopRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request params
export const stopRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN" /* required */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new StopRecommenderCommand(stopRecommenderParam)
    );
    console.log("Success", response);
```

```
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize

Si vous utilisez une recette personnalisée, après avoir importé les données, vous êtes prêt à créer les ressources personnalisées pour la formation et le déploiement des modèles Amazon Personalize. Vous utilisez ces ressources pour obtenir des recommandations. Lorsque vous créez des ressources personnalisées, vous devez effectuer les opérations suivantes :

1. Création et configuration d'une solution : personnalisez les paramètres de la solution et les hyperparamètres spécifiques à la recette afin que le modèle réponde aux besoins spécifiques de votre entreprise. Par défaut, les nouvelles versions de solution utilisent l'entraînement automatique pour créer des versions de solution à une fréquence configurable. La fréquence par défaut est tous les 7 jours. Pour plus d'informations sur la configuration d'une solution, consultezConfiguration d'une solution. Pour plus d'informations sur les recettes personnalisées dans Amazon Personalize, consultezChoisir une recette.
2. Créer une version de solution (pour les solutions qui n'utilisent pas d'entraînement automatique) : pour les solutions qui utilisent l'entraînement automatique, la création de version de solution démarre automatiquement une fois que votre solution est active. Pour les solutions qui utilisent une formation manuelle, vous créez manuellement une version de solution. La version de la solution génère des recommandations Amazon Personalize ou des segments d'utilisateurs. Pour plus d'informations sur la création manuelle d'une version de solution, consultezCréation manuelle d'une version de solution. Pour arrêter la création d'une version de solution, accédez à la page de détails de la version de solution et choisissez Arrêter. Pour de plus amples informations, veuillez consulter Arrêt de la création d'une version de solution.
3. Évaluer la version de la solution : utilisez les indicateurs générés par Amazon Personalize à partir de la nouvelle version de la solution pour évaluer les performances du modèle. Consultez Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques.
4. Déployez la version de la solution avec une campagne (uniquement pour les recommandations en temps réel) : créez une campagne pour déployer la version de votre solution. Vous utilisez la campagne lorsque vous demandez des recommandations en temps réel. Si vous recevez des recommandations par lots, vous n'avez pas besoin de créer de campagne. Pour de plus amples informations, veuillez consulter Déploiement d'une version de la solution Amazon Personalize avec une campagne. Si vous souhaitez modifier les paramètres d'une campagne existante, par exemple en activant les métadonnées dans les recommandations, vous devez mettre à jour votre

campagne. Pour de plus amples informations, veuillez consulter Mettre à jour la configuration d'une campagne Amazon Personalize.

Rubriques

- Configuration d'une solution personnalisée dans Amazon Personalize
- Mettre à jour une solution pour modifier sa configuration d'entraînement automatique
- Création manuelle d'une version de solution
- Arrêt de la création d'une version de solution
- Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques
- Déploiement d'une version de la solution Amazon Personalize avec une campagne
- Mettre à jour la configuration d'une campagne Amazon Personalize

Configuration d'une solution personnalisée dans Amazon Personalize

Une fois l'importation des données terminée, vous êtes prêt à créer une solution. Une solution fait référence à la combinaison d'une recette Amazon Personalize, de paramètres d'entraînement personnalisés et d'une ou plusieurs versions de solution. Une version de solution fait référence à un modèle d'apprentissage automatique entraîné.

Par défaut, toutes les nouvelles solutions utilisent une formation automatique pour créer une nouvelle version de solution tous les 7 jours. L'entraînement automatique a lieu uniquement si vous avez importé des données d'interaction en masse ou en temps réel depuis le dernier entraînement. Cela inclut les interactions entre les éléments ou, pour les solutions utilisant la Next-Best-Action recette, les données relatives aux interactions entre les actions. L'entraînement automatique se poursuit jusqu'à ce que vous supprimiez la solution. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique.

Si vous avez une solution existante, vous pouvez utiliser la console Amazon Personalize pour cloner la solution. Lorsque vous clonez une solution, vous pouvez utiliser la configuration de la solution existante comme point de départ, comme la recette et les hyperparamètres, et apporter des modifications. Pour de plus amples informations, veuillez consulter Clonage d'une solution (console).

Vous pouvez créer et configurer une solution à l'aide de la console, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Après avoir créé une solution, vous pouvez consulter les détails de sa

configuration sur la page de détails de la solution de la console Amazon Personalize, ou lors de l'DescribeSolutionopération.

Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Rubriques

- Création d'une solution
- Configuration de l'entraînement automatique
- Configuration des colonnes utilisées lors de l'entraînement
- Optimisation d'une solution pour un objectif supplémentaire
- Optimisation d'une solution avec configuration d'événements
- Hyperparamètres et HPO
- Choix des données d'interaction entre les objets utilisées pour l'entraînement
- Clonage d'une solution (console)

Création d'une solution

Vous pouvez créer une solution personnalisée avec la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Vous trouverez ci-dessous des étapes détaillées pour créer une solution avec la console Amazon Personalize et des exemples de code qui montrent comment créer une solution avec uniquement les champs obligatoires.

Rubriques

- Création d'une solution (console)
- Création d'une solution (AWS CLI)
- Création d'une solution (AWS SDKs)

Création d'une solution (console)

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Pour créer une solution dans la console, choisissez votre groupe de jeux de données, puis spécifiez un nom de solution, une recette et une configuration d'entraînement facultative.

Pour configurer une solution (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Sur la page Vue d'ensemble, pour l'étape 3, effectuez l'une des opérations suivantes :
   - Si vous avez créé un groupe de jeux de données de domaine, choisissez Utiliser des ressources personnalisées, puis sélectionnez Créer des solutions.
   - Si vous avez créé un groupe de jeux de données personnalisé, choisissez Create solutions.
4. Pour Solution name (Nom de la solution), spécifiez un nom pour votre solution.
5. Dans Type de solution, choisissez le type de solution que vous souhaitez créer. Le type que vous choisissez détermine les recettes disponibles.
   - Choisissez Recommandation d'articles pour obtenir des recommandations d'articles pour vos utilisateurs. Par exemple, des recommandations de films personnalisées.
   - Choisissez Recommandation d'action pour obtenir des recommandations d'action pour vos utilisateurs. Par exemple, générez la meilleure action suivante pour un utilisateur, telle que télécharger votre application.
   - Choisissez la segmentation des utilisateurs pour obtenir des segments d'utilisateurs (groupes d'utilisateurs) basés sur les données de vos articles.
6. Pour Recette, choisissez une recette (voirChoisir une recette).

7. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
8. Choisissez Suivant.
9. Sur la page de configuration de la formation, personnalisez la solution pour répondre aux besoins de votre entreprise.

   - Dans Entraînement automatique, indiquez si la solution utilise l'entraînement automatique. Si vous utilisez l'entraînement automatique, vous pouvez modifier le`Automatic training frequency`. La fréquence d'entraînement par défaut est tous les 7 jours.

     Nous vous recommandons d'utiliser l'entraînement automatique. Cela vous permet de maintenir plus facilement la pertinence des recommandations. La fréquence de vos formations dépend des besoins de votre entreprise, de la recette que vous utilisez et de la fréquence à laquelle vous importez des données. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique. Pour plus d'informations sur le maintien de la pertinence, voirMaintien de la pertinence des recommandations.
   - Dans Configuration des hyperparamètres, configurez les options des hyperparamètres en fonction de votre recette et des besoins de votre entreprise. Différentes recettes utilisent différents hyperparamètres. Pour les hyperparamètres mis à votre disposition, consultez les recettes individuelles dansChoisir une recette.
   - Dans Columns for training, si votre recette génère des recommandations d'articles ou des segments d'utilisateurs, choisissez éventuellement les colonnes prises en compte par Amazon Personalize lors de la création de versions de solution. Pour de plus amples informations, veuillez consulter Configuration des colonnes utilisées lors de l'entraînement.
   - Dans Configuration des événements, si votre ensemble de données d'interactions avec les articles contient les colonnes EVENT_TYPE ou EVENT_TYPE et EVENT_VALUE, utilisez éventuellement les champs Type d'événement et Seuil de valeur d'événement pour choisir les données d'interactions entre articles utilisées par Amazon Personalize lors de la formation du modèle. Pour de plus amples informations, veuillez consulter Choix des données d'interaction entre les objets utilisées pour l'entraînement.

     Si vous avez plusieurs types d'événements et que vous utilisez la recette User-Personalization-v 2 ou Personalized-Ranking-v 2 recettes, vous pouvez également spécifier des pondérations différentes pour les différents types. Par exemple, vous pouvez configurer une solution pour donner plus de poids aux événements d'achat qu'aux événements de

clics. Pour de plus amples informations, veuillez consulter Optimisation d'une solution avec configuration d'événements.

- Si vous utilisez la Recette de classement personnalisé recette Recette de personnalisation par l'utilisateur ou, vous pouvez éventuellement spécifier un objectif et choisir une sensibilité d'objectif afin d'optimiser votre solution pour un objectif en plus de la pertinence. La sensibilité objective configure la façon dont Amazon Personalize équilibre les recommandations d'articles en fonction de votre objectif par rapport à la pertinence grâce aux données d'interactions. Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.

10. Choisissez Next et passez en revue les détails de la solution. Vous ne pouvez pas modifier la configuration de votre solution après l'avoir créée.
11. Choisissez Create solution (Créer unes solution). Une fois que vous avez créé une solution, Amazon Personalize commence à créer votre première version de solution en moins d'une heure. Lorsque la formation commence, vous pouvez la suivre dans la section Versions de la solution sur la page de détails de votre solution. Les versions de solution créées automatiquement ont le type d'entraînement AUTOMATIQUE.

    Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

    - Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.
    - Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Création d'une solution (AWS CLI)

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est

active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Pour créer une solution avec le AWS CLI, utilisez la `create-solution` commande. Cette commande utilise l'opération CreateSolution API. Le code suivant explique comment créer une solution utilisant l'entraînement automatique. Il crée automatiquement une nouvelle version de la solution tous les cinq jours.

Pour utiliser le code, mettez-le à jour pour donner un nom à la solution, spécifiez l'Amazon Resource Name (ARN) de votre groupe de données, modifiez éventuellement la fréquence d'entraînement et spécifiez l'ARN de la recette à utiliser. Pour plus d'informations sur les recettes, consultez Choisir une recette.

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--perform-auto-training \
--solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5
 days)\"}}"
```

- Nous vous recommandons d'utiliser l'entraînement automatique. Cela vous permet de maintenir et d'améliorer plus facilement la pertinence des recommandations. Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. La fréquence d'entraînement par défaut est tous les 7 jours. La fréquence de vos formations dépend des besoins de votre entreprise, de la recette que vous utilisez et de la fréquence à laquelle vous importez des données. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique.
- En fonction de votre recette, vous pouvez modifier le code pour configurer des propriétés et des hyperparamètres spécifiques à la recette (voirHyperparamètres et HPO), configurer les colonnes utilisées pour l'entraînement (voirConfiguration des colonnes utilisées lors de l'entraînement (AWS CLI)) ou filtrer les données d'interactions entre éléments utilisées pour l'entraînement (voirChoix des données d'interaction entre les objets utilisées pour l'entraînement).
- Si vous utilisez la Recette de classement personnalisé recette Recette de personnalisation par l'utilisateur ou, vous pouvez optimiser votre solution en fonction d'un objectif, en plus de sa pertinence. Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.

Après avoir créé la solution, enregistrez l'ARN de la solution pour une utilisation future. Avec l'entraînement automatique, la création de versions de solution commence dans l'heure qui suit l'activation de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore le premier entraînement automatique. Après le début de la formation, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution avec le fonctionnement de l'ListSolutionVersionsAPI. Pour obtenir son statut, utilisez l'opération DescribeSolutionVersionAPI.

Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

- Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.
- Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Création d'une solution (AWS SDKs)

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Pour créer une solution avec AWS SDKs, utilisez l'opération CreateSolution API. Le code suivant explique comment créer une solution utilisant l'entraînement automatique. Il crée automatiquement une nouvelle version de la solution tous les cinq jours.

Pour utiliser le code, mettez-le à jour pour donner un nom à la solution, spécifiez l'Amazon Resource Name (ARN) de votre groupe de données, modifiez éventuellement la fréquence d'entraînement et spécifiez l'ARN de la recette que vous souhaitez utiliser. Pour plus d'informations sur les recettes, consultez Choisir une recette.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
  name = 'solution name',
  recipeArn = 'recipe ARN',
  datasetGroupArn = 'dataset group ARN',
  performAutoTraining = True,
  solutionConfig = {
    "autoTrainingConfig": {
      "schedulingExpression": "rate(5 days)"
    }
  }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for JavaScript v3

```

import {
  CreateSolutionCommand,
  PersonalizeClient,
} from "@aws-sdk/client-personalize";

// create client
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the solution parameters
export const solutionParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  name: "SOLUTION_NAME" /* required */,
  performAutoTraining: true /* optional, default is true */,
  solutionConfig: {
    autoTrainingConfig: {
      schedulingExpression:
        "rate(5 days)" /* optional, default is every 7 days */,
    },
  },
```

```
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionCommand(solutionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Nous vous recommandons d'utiliser l'entraînement automatique. Cela vous permet de maintenir et d'améliorer plus facilement la pertinence des recommandations. Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. La fréquence d'entraînement par défaut est tous les 7 jours. La fréquence de vos formations dépend des besoins de votre entreprise, de la recette que vous utilisez et de la fréquence à laquelle vous importez des données. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique.
- En fonction de votre recette, vous pouvez modifier le code pour configurer des propriétés et des hyperparamètres spécifiques à la recette (voirHyperparamètres et HPO), configurer les colonnes utilisées pour l'entraînement (voirConfiguration des colonnes utilisées lors de l'entraînement (AWS SDK)) ou filtrer les données d'interactions entre éléments utilisées pour l'entraînement (voirChoix des données d'interaction entre les objets utilisées pour l'entraînement).
- Si vous utilisez la Recette de classement personnalisé recette Recette de personnalisation par l'utilisateur ou, vous pouvez optimiser votre solution en fonction d'un objectif, en plus de sa pertinence. Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.

Après avoir créé la solution, enregistrez l'ARN de la solution pour une utilisation future. Avec l'entraînement automatique, la création de versions de solution commence dans l'heure qui suit l'activation de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore le premier entraînement automatique. Après le début de la formation, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution avec le fonctionnement de l'ListSolutionVersionsAPI. Pour obtenir son statut, utilisez l'opération DescribeSolutionVersionAPI.

Vous pouvez utiliser le code Python suivant pour attendre le début de l'entraînement automatique. La `wait_for_training_to_start` méthode renvoie l'ARN de la première version de la solution.

```
import time
import boto3


def wait_for_training_to_start(new_solution_arn):
    max_time = time.time() + 3 * 60 * 60    # 3 hours
    while time.time() < max_time:
        list_solution_versions_response = personalize.list_solution_versions(
            solutionArn=new_solution_arn
        )
        solution_versions = list_solution_versions_response.get('solutionVersions', [])
        if solution_versions:
            new_solution_version_arn = solution_versions[0]['solutionVersionArn']
            print(f"Solution version ARN: {new_solution_version_arn}")
            return new_solution_version_arn
        else:
            print(f"Training hasn't started yet. Training will start within the next
 hour.")
            time.sleep(60)


personalize = boto3.client('personalize')

solution_arn = "solution_arn"
solution_version_arn = wait_for_training_to_start(solution_arn)
```

Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

- Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.
- Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Configuration de l'entraînement automatique

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Lorsque vous créez une solution, vous pouvez configurer si la solution utilise l'entraînement automatique. Vous pouvez également configurer la fréquence d'entraînement. Par exemple, vous pouvez configurer la solution pour créer une nouvelle version de solution tous les cinq jours.

Par défaut, toutes les nouvelles solutions utilisent une formation automatique pour créer une nouvelle version de solution tous les 7 jours. L'entraînement automatique a lieu uniquement si vous avez importé des données d'interaction en masse ou en temps réel depuis le dernier entraînement. Cela inclut les interactions entre les éléments ou, pour les solutions utilisant la Next-Best-Action recette, les données relatives aux interactions entre les actions. L'entraînement automatique se poursuit jusqu'à ce que vous supprimiez la solution.

Nous vous recommandons d'utiliser l'entraînement automatique. Cela facilite la maintenance de votre solution. Cela supprime la formation manuelle requise pour que la solution puisse tirer parti de vos données les plus récentes. Sans formation automatique, vous devez créer manuellement de nouvelles versions de solution pour que la solution puisse tirer des leçons de vos données les plus récentes. Cela peut entraîner des recommandations périmées et un taux de conversion plus faible. Pour plus d'informations sur la mise à jour des recommandations d'Amazon Personalize, consultezMaintien de la pertinence des recommandations.

Vous pouvez configurer l'entraînement automatique avec la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Pour connaître les étapes de configuration de l'entraînement automatique avec la console, voirCréation d'une solution (console).

Après avoir créé la solution, enregistrez l'ARN de la solution pour une utilisation future. Avec l'entraînement automatique, la création de versions de solution commence dans l'heure qui suit l'activation de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore le premier entraînement automatique. Après le début de la formation, vous pouvez

obtenir le nom de ressource Amazon (ARN) de la version de la solution avec le fonctionnement de l'ListSolutionVersionsAPI. Pour obtenir son statut, utilisez l'opération DescribeSolutionVersionAPI.

Rubriques

- Directives et exigences
- Configuration de l'entraînement automatique (AWS CLI)
- Configuration de l'entraînement automatique (SDKs)

Directives et exigences

Les directives et les exigences relatives à l'entraînement automatique sont les suivantes :

- L'entraînement automatique a lieu uniquement si vous avez importé des données d'interaction en masse ou en temps réel depuis le dernier entraînement. Cela inclut les interactions entre les éléments ou, pour les solutions utilisant la Next-Best-Action recette, les données relatives aux interactions entre les actions.
- Chaque formation prend en compte toutes les données de votre groupe de jeux de données que vous incluez dans la formation. Pour plus d'informations sur la configuration des colonnes utilisées lors de l'entraînement, consultezConfiguration des colonnes utilisées lors de l'entraînement.
- Vous pouvez toujours créer des versions de solution manuellement.
- L'entraînement automatique démarre dans l'heure qui suit l'activation de votre solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore le premier entraînement automatique.
- La planification de la formation est basée sur la date de début de la formation. Par exemple, si votre première version de solution commence à s'entraîner à 19 h 00 et que vous utilisez une formation hebdomadaire, la version suivante débutera l'entraînement une semaine plus tard à 19 h 00.
- Pour toutes les recettes, nous recommandons au moins une fréquence d'entraînement hebdomadaire. Vous pouvez définir une fréquence d'entraînement comprise entre 1 et 30 jours. La valeur par défaut est tous les 7 jours.
  - Si vous utilisez User-Personalization-v 2, Personnalisation utilisateur, ou Next-Best-Action, la solution est automatiquement mise à jour pour prendre en compte les nouveaux éléments ou actions à recommander. Les mises à jour automatiques ne sont pas la même chose que l'entraînement automatique. Une mise à jour automatique ne crée pas une version entièrement nouvelle de la solution, et le modèle ne tire aucune leçon de vos données les plus récentes. Pour

maintenir votre solution, votre fréquence d'entraînement doit toujours être d'au moins une fois par semaine. Pour plus d'informations sur les mises à jour automatiques, y compris les directives et exigences supplémentaires, voirMises à jour automatiques.

- Si vous utilisez Trending-Now, Amazon Personalize identifie automatiquement les éléments les plus populaires dans vos données d'interactions sur un intervalle de temps configurable. Trending-Now peut recommander des éléments ajoutés depuis le dernier entraînement par le biais de données d'interactions en masse ou en streaming. Votre fréquence d'entraînement doit toujours être d'au moins une fois par semaine. Pour de plus amples informations, veuillez consulter Recette Trending-Now.
- Si vous n'utilisez pas de recette avec mises à jour automatiques ou la recette Trending-Now, Amazon Personalize prend en compte les nouveaux articles pour les recommandations uniquement après la prochaine formation. Par exemple, si vous utilisez la recette Similar-Items et que vous ajoutez de nouveaux articles tous les jours, vous devrez utiliser une fréquence d'entraînement automatique quotidienne pour que ces éléments apparaissent dans les recommandations le jour même.

Configuration de l'entraînement automatique (AWS CLI)

Le code suivant explique comment créer une solution qui crée automatiquement une version de solution tous les cinq jours. Pour désactiver l'entraînement automatique, réglez `perform-auto-training` sur`false`.

Pour modifier la fréquence d'entraînement, vous pouvez modifier le `schedulingExpression` dans le`autoTrainingConfig`. L'expression doit être au `rate(`*`value unit`*`)` format. Pour la valeur, spécifiez un nombre compris entre 1 et 30. Pour l'unité, spécifiez `day` ou`days`.

Pour une explication complète de la `create-solution` commande, voirCréation d'une solution (AWS CLI).

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--perform-auto-training \
--solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5
 days)\"}}"
```

Configuration de l'entraînement automatique (SDKs)

Le code suivant explique comment créer une solution avec entraînement automatique à l'aide du AWS SDKs. La solution crée automatiquement une version de la solution tous les cinq jours. Pour désactiver l'entraînement automatique, réglez `performAutoTraining` sur`false`.

Pour modifier la fréquence d'entraînement, vous pouvez modifier le `schedulingExpression` dans le`autoTrainingConfig`. L'expression doit être au `rate(`*`value unit`*`)` format. Pour la valeur, spécifiez un nombre compris entre 1 et 30. Pour l'unité, spécifiez `day` ou`days`.

Pour une explication complète du fonctionnement de l' CreateSolution API, consultezCréation d'une solution (AWS SDKs).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
  name = 'solution name',
  recipeArn = 'recipe ARN',
  datasetGroupArn = 'dataset group ARN',
  performAutoTraining = True,
  solutionConfig = {
    "autoTrainingConfig": {
      "schedulingExpression": "rate(5 days)"
    }
  }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for JavaScript v3

```

import {
  CreateSolutionCommand,
  PersonalizeClient,
} from "@aws-sdk/client-personalize";

// create client
```

```
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the solution parameters
export const solutionParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  name: "SOLUTION_NAME" /* required */,
  performAutoTraining: true /* optional, default is true */,
  solutionConfig: {
    autoTrainingConfig: {
      schedulingExpression:
        "rate(5 days)" /* optional, default is every 7 days */,
    },
  },
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionCommand(solutionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Vous pouvez utiliser le code Python suivant pour attendre le début de l'entraînement automatique. La `wait_for_training_to_start` méthode renvoie l'ARN de la première version de la solution.

```
import time
import boto3

def wait_for_training_to_start(new_solution_arn):
    max_time = time.time() + 3 * 60 * 60    # 3 hours
    while time.time() < max_time:
        list_solution_versions_response = personalize.list_solution_versions(
            solutionArn=new_solution_arn
        )
        solution_versions = list_solution_versions_response.get('solutionVersions', [])
```

```
        if solution_versions:
            new_solution_version_arn = solution_versions[0]['solutionVersionArn']
            print(f"Solution version ARN: {new_solution_version_arn}")
            return new_solution_version_arn
        else:
            print(f"Training hasn't started yet. Training will start within the next
 hour.")
            time.sleep(60)


personalize = boto3.client('personalize')

solution_arn = "solution_arn"
solution_version_arn = wait_for_training_to_start(solution_arn)
```

Configuration des colonnes utilisées lors de l'entraînement

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Si votre recette génère des recommandations d'articles ou des segments d'utilisateurs, vous pouvez modifier les colonnes prises en compte par Amazon Personalize lors de la création d'une version de solution (formation d'un modèle).

Vous pouvez modifier les colonnes utilisées lors de l'entraînement afin de contrôler les données utilisées par Amazon Personalize lors de la formation d'un modèle (création d'une version de solution). Vous pouvez le faire pour expérimenter différentes combinaisons de données d'entraînement. Vous pouvez également exclure des colonnes dépourvues de données significatives. Par exemple, il peut y avoir une colonne que vous souhaitez utiliser uniquement pour filtrer les recommandations. Vous pouvez exclure cette colonne de la formation et Amazon Personalize ne la prend en compte que lors du filtrage.

Vous ne pouvez pas exclure les colonnes EVENT_TYPE. Par défaut, Amazon Personalize utilise toutes les colonnes qui peuvent être utilisées lors de la formation. Les données suivantes sont toujours exclues de la formation :

- Colonnes avec le type de données booléen
- Données sur les impressions
- Champs de chaîne personnalisés qui ne sont ni catégoriques ni textuels

Vous ne pouvez pas inclure de données sur les impressions dans la formation, mais si votre cas d'utilisation ou votre recette les utilise, Amazon Personalize utilise les données d'impressions pour guider l'exploration lorsque vous recevez des recommandations.

Si vous avez déjà créé une solution et que vous souhaitez modifier les colonnes qu'elle utilise lors de l'entraînement, vous pouvez cloner la solution. Lorsque vous clonez une solution, vous pouvez utiliser la configuration de la solution existante comme point de départ, comme la recette et les hyperparamètres, et apporter les modifications nécessaires. Pour plus d’informations, consultez Clonage d'une solution (console).

Vous pouvez configurer les colonnes qu'Amazon Personalize utilise lors de l'entraînement à l'aide de la console Amazon Personalize AWS Command Line Interface (AWS CLI) ou du AWS SDK. Pour plus d'informations sur le choix des colonnes avec la console Amazon Personalize, consultez les étapes de configuration avancées décrites dansCréation d'une solution (console). Après avoir créé une solution, vous pouvez consulter les colonnes qu'elle utilise sur la page de détails de la solution de la console Amazon Personalize, ou lors de l'DescribeSolutionopération.

Rubriques

- Configuration des colonnes utilisées lors de l'entraînement (AWS CLI)
- Configuration des colonnes utilisées lors de l'entraînement (AWS SDK)

Configuration des colonnes utilisées lors de l'entraînement (AWS CLI)

Pour exclure des colonnes de l'apprentissage, fournissez l'`excludedDatasetColumns`objet dans le `trainingDataConfig` cadre de la configuration de la solution. Pour chaque clé, indiquez le type de jeu de données. Pour chaque valeur, fournissez la liste des colonnes à exclure. Le code suivant montre comment exclure des colonnes de l'entraînement lorsque vous créez une solution avec le AWS CLI.

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--solution-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

Configuration des colonnes utilisées lors de l'entraînement (AWS SDK)

Pour exclure des colonnes de l'apprentissage, fournissez l'`excludedDatasetColumns`objet dans le `trainingDataConfig` cadre de la configuration de la solution. Pour chaque clé, indiquez le type de jeu de données. Pour chaque valeur, fournissez la liste des colonnes à exclure. Le code suivant montre comment exclure des colonnes de l'entraînement lorsque vous créez une solution avec le AWS SDK pour Python (Boto3).

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
  name = 'solution name',
  recipeArn = 'recipe ARN',
  datasetGroupArn = 'dataset group ARN',
  solutionConfig = {
    "trainingDataConfig": {
      "excludedDatasetColumns": {
          "datasetType": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

Optimisation d'une solution pour un objectif supplémentaire

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est

active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Si vous utilisez la recette de personnalisation utilisateur ou la recette de classement personnalisé, vous pouvez optimiser une solution Amazon Personalize en fonction d'un objectif en plus d'une pertinence maximale, comme la maximisation des revenus.

Avec les recettes de recommandation d'articles, l'objectif principal d'Amazon Personalize est de prévoir les articles les plus pertinents pour vos utilisateurs sur la base de données historiques et en temps réel sur les interactions entre articles. Il s'agit des éléments avec lesquels vos utilisateurs sont les plus susceptibles d'interagir (par exemple, les éléments sur lesquels ils seront le plus susceptibles de cliquer). Si vous avez un objectif supplémentaire, tel que la maximisation des minutes de streaming ou l'augmentation des revenus, vous pouvez créer une solution qui génère des recommandations basées à la fois sur la pertinence et sur votre objectif.

Pour optimiser une solution pour un objectif supplémentaire, créez une nouvelle solution à l'aide de la recette de personnalisation utilisateur ou de la recette de classement personnalisé et choisissez la colonne de métadonnées numériques de votre jeu de données d'articles qui est liée à votre objectif. Lors de la génération de recommandations, Amazon Personalize accorde plus d'importance aux éléments présentant des valeurs plus élevées pour cette colonne de données. Par exemple, vous pouvez choisir une colonne VIDEO_LENGTH pour maximiser les minutes de diffusion ou une colonne PRICE pour maximiser les revenus.

Vous pouvez utiliser la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Pour plus d'informations sur l'utilisation de la console Amazon Personalize, consultezCréation d'une solution (console).

Rubriques

- Directives et exigences
- Équilibrer l'objectif, l'accent et la pertinence
- Mesurer les performances d'optimisation
- Optimisation d'une solution (AWS CLI)
- Optimisation d'une solution (AWS SDKs)
- Exemple de carnet Jupyter

Directives et exigences

Les exigences objectives sont les suivantes :

- Vous ne pouvez choisir qu'une seule colonne pour votre objectif.
- La colonne doit avoir un type numérique dans votre schéma.
- La colonne ne peut pas avoir de `null` type dans votre schéma.

Pour plus d'informations sur les schémas et les types de données, consultezCréation de fichiers JSON pour les schémas Amazon Personalize.

Équilibrer l'objectif, l'accent et la pertinence

Il peut y avoir un compromis lorsque vous recommandez des articles en fonction de votre objectif plutôt que de leur pertinence. Par exemple, si vous souhaitez augmenter vos revenus par le biais de recommandations, les recommandations portant uniquement sur des articles coûteux peuvent rendre les articles moins pertinents pour vos utilisateurs et réduire l'engagement et le taux de conversion des utilisateurs.

Pour configurer l'équilibre entre la pertinence et votre objectif, choisissez l'un des niveaux de sensibilité d'objectif suivants lorsque vous créez la solution :

- Désactivé : Amazon Personalize utilise principalement les données relatives aux interactions avec les articles pour prévoir les articles les plus pertinents pour votre utilisateur.
- Faible : Amazon Personalize met moins l'accent sur votre objectif. La pertinence des données relatives aux interactions entre les articles est plus importante.
- Médium : Amazon Personalize met autant l'accent sur votre objectif que sur votre pertinence par le biais des données relatives aux interactions avec les articles.
- Élevé : Amazon Personalize met davantage l'accent sur votre objectif. La pertinence des données relatives aux interactions entre les articles est moins importante.

Mesurer les performances d'optimisation

Lorsque vous créez une version de solution (entraînez un modèle) pour une solution avec un objectif d'optimisation, Amazon Personalize génère une `average_rewards_at_k` métrique. Le score de `average_rewards_at_k` indique dans quelle mesure la version de la solution permet d'atteindre

votre objectif. Pour calculer cette statistique, Amazon Personalize calcule les récompenses pour chaque utilisateur comme suit :

```
rewards_per_user = total rewards from the user's interactions with their
top 25 reward generating recommendations / total rewards from the user's
interactions with recommendations
```

La valeur finale `average_rewards_at_k` est la moyenne de toutes les valeurs `rewards_per_user` normalisées pour être une valeur décimale inférieure ou égale à 1 et supérieure à 0. Plus la valeur est proche de 1, plus vous pouvez vous attendre à des gains en moyenne par utilisateur grâce aux recommandations.

Par exemple, si votre objectif est de maximiser les revenus générés par les clics, Amazon Personalize calcule le score de chaque utilisateur en divisant le chiffre d'affaires total généré par les articles sur lesquels l'utilisateur a cliqué parmi ses 25 recommandations les plus chères par les revenus provenant de tous les articles recommandés sur lesquels l'utilisateur a cliqué. Amazon Personalize renvoie ensuite une moyenne normalisée de tous les scores des utilisateurs. Plus la valeur `average_rewards_at_k` est proche de 1, plus vous pouvez vous attendre à générer en moyenne des revenus par utilisateur grâce aux recommandations.

Pour plus d'informations sur la génération de métriques, consultezÉvaluation d'une version de la solution Amazon Personalize à l'aide de métriques.

Optimisation d'une solution (AWS CLI)

Vous ne pouvez optimiser un objectif qu'à l'aide de la recette de personnalisation par l'utilisateur ou de classement personnalisé. Pour optimiser une solution pour un objectif supplémentaire à l'aide du AWS CLI, créez une nouvelle solution et spécifiez les détails de votre objectif à l'aide de la `optimizationObjective` clé de l'`solutionConfig`objet. `optimizationObjective`comporte les champs suivants :

- `itemAttribute`: Spécifiez le nom de la colonne de métadonnées numériques du jeu de données Items associée à votre objectif.
- `objectiveSensitivity`: Spécifiez le niveau d'importance que la solution met sur votre objectif lors de la génération de recommandations. Le niveau de sensibilité des objectifs configure la manière dont Amazon Personalize équilibre les recommandations d'articles en fonction de votre objectif par rapport à leur pertinence grâce aux données d'interaction entre les articles. Ils `objectiveSensitivity` peuvent être `OFF` FAIBLES `MEDIUM` ou`HIGH`. Pour de plus amples informations, veuillez consulter Équilibrer l'objectif, l'accent et la pertinence.

Voici un exemple de la commande `create-solution` AWS CLI . Remplacez les `recipe arn` valeurs `solution namedataset group arn`, et par les vôtres.

Pour`optimizationObjective`, remplacez-le `COLUMN_NAME` par le nom de colonne de métadonnées numériques du jeu de données Items associé à votre objectif. Pour`objectiveSensitivity`, spécifiez OFF, LOW, MEDIUM ou HIGH.

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group arn \
--recipe-arn recipe arn \
--solution-config "{\"optimizationObjective\":{\"itemAttribute\":\"COLUMN_NAME\",
\"objectiveSensitivity\":\"MEDIUM\"}}"
```

Lorsque votre solution est prête, créez une nouvelle version de solution (pour un exemple de commande, voirCréation d'une solution (AWS CLI)). Une fois que vous avez créé une version de solution, vous pouvez visualiser les performances d'optimisation à l'aide des métriques de version de solution. Consultez Mesurer les performances d'optimisation.

Optimisation d'une solution (AWS SDKs)

Vous ne pouvez optimiser un objectif qu'à l'aide de la recette de personnalisation par l'utilisateur ou de classement personnalisé.

Pour optimiser une solution pour un objectif supplémentaire à l'aide du AWS SDKs, créez une nouvelle solution et spécifiez les détails de votre objectif à l'aide de la `optimizationObjective` clé de l'`solutionConfig`objet correspondant à la solution. `optimizationObjective`comporte les champs suivants :

- `itemAttribute`: Spécifiez le nom de la colonne de métadonnées numériques du jeu de données Items du groupe de données qui se rapporte à votre objectif.
- `objectiveSensitivity`: Spécifiez le niveau d'importance que la solution met sur votre objectif lors de la génération de recommandations. Le niveau de sensibilité des objectifs configure la manière dont Amazon Personalize équilibre les recommandations d'articles en fonction de votre objectif par rapport à leur pertinence grâce aux données d'interaction entre les articles. Ils `objectiveSensitivity` peuvent être `OFFLOW`, `MEDIUM` ou`HIGH`. Pour de plus amples informations, veuillez consulter Équilibrer l'objectif, l'accent et la pertinence.

Utilisez le code suivant pour créer une solution avec un objectif supplémentaire avec le AWS SDK pour Python (Boto3) ou le AWS SDK for Java 2.x.

Lorsque votre solution est prête, créez une nouvelle version de la solution (par exemple, voir le codeCréation d'une version de solution (AWS SDKs)). Une fois que vous avez créé une version de solution, vous pouvez visualiser les performances d'optimisation à l'aide des métriques de version de solution. Consultez Mesurer les performances d'optimisation.

SDK for Python (Boto3)

Pour créer une solution optimisée pour un objectif supplémentaire, appliquez la `create_solution` méthode suivante. Remplacez les `recipe arn` valeurs `solution namedataset group arn`, et par les vôtres.

Pour`optimizationObjective`, remplacez-le `COLUMN_NAME` par le nom de colonne de métadonnées numériques du jeu de données Items associé à votre objectif. Pour`objectiveSensitivity`, spécifiez OFF, LOW, MEDIUM ou HIGH.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name= 'solution name',
    recipeArn = 'recipe arn',
    datasetGroupArn = 'dataset group arn',
    solutionConfig = {
        "optimizationObjective": {
            "itemAttribute": "COLUMN_NAME",
            "objectiveSensitivity": "MEDIUM"
        }
    }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for Java 2.x

Pour créer une solution optimisée pour un objectif supplémentaire, utilisez la `createPersonalizeSolution` méthode suivante et transmettez les paramètres suivants : un client du service Amazon Personalize, le nom de ressource Amazon (ARN) du groupe de

données, le nom de la solution, l'ARN de la recette, l'attribut de l'article et le niveau de sensibilité de l'objectif.

```
public static String createPersonalizeSolution(PersonalizeClient personalizeClient,
                                                String datasetGroupArn,
                                                String solutionName,
                                                String recipeArn,
                                                String itemAttribute,
                                                String objectiveSensitivity) {

    try {
        OptimizationObjective optimizationObjective =
 OptimizationObjective.builder()
            .itemAttribute(itemAttribute)
            .objectiveSensitivity(objectiveSensitivity)
            .build();

        SolutionConfig solutionConfig = SolutionConfig.builder()
            .optimizationObjective(optimizationObjective)
            .build();

        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .solutionConfig(solutionConfig)
            .build();

        CreateSolutionResponse solutionResponse =
 personalizeClient.createSolution(solutionRequest);

        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateSolutionCommand, PersonalizeClient } from
```

```
  "@aws-sdk/client-personalize";

// create the personalizeClient
const personalizeClient = new PersonalizeClient({ region: "REGION"});

// set the solution parameters.
export const createSolutionParam = {
  datasetGroupArn: 'DATASET_GROUP_ARN',             /* required */
  recipeArn: 'RECIPE_ARN',                          /* required */
  name: 'NAME',                                     /* required */
  solutionConfig: {
    optimizationObjective: {
      itemAttribute: "COLUMN_NAME",           /* specify the numerical column from
 the Items dataset related to your objective */
      objectiveSensitivity: "MEDIUM"          /* specify OFF, LOW, MEDIUM, or HIGH
 */
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 CreateSolutionCommand(createSolutionParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Exemple de carnet Jupyter

Pour un exemple de bloc-notes Jupyter qui montre comment créer une solution optimisée pour des métadonnées d'article supplémentaires basées sur un objectif, consultez le dossier objective_optimization du référentiel d'échantillons Amazon Personalize GitHub

Optimisation d'une solution avec configuration d'événements

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Si vous utilisez la recette User-Personalization-v 2 ou Personalized-Ranking-v 2 recettes, vous pouvez optimiser une solution Amazon Personalize avec une configuration d'événements.

Avec les recettes de recommandation d'articles, l'objectif principal d'Amazon Personalize est de prévoir les articles les plus pertinents pour vos utilisateurs sur la base de données historiques et en temps réel sur les interactions entre articles. Cependant, l'interaction peut contenir des informations supplémentaires, par exemple si un utilisateur a cliqué ou acheté un certain article. Vous pouvez l'enregistrer en enregistrant le type de l'événement (Données relatives au type d'événement et à la valeur de l'événement). Lorsque vous configurez une solution, celle-ci peut attribuer une pondération différente aux différents types d'événements d'interaction. Par exemple, vous pouvez configurer une solution pour donner plus de poids aux `purchase` événements qu'aux `click` événements.

Pour qu'une solution attribue des pondérations différentes aux différents types d'événements, vous devez spécifier les types d'événements et les poids correspondants dans la configuration des événements de la solution. En outre, vous pouvez définir un seuil de valeur d'événement pour exclure les interactions dont la valeur d'événement est inférieure à ce seuil. Par exemple, si vos données EVENT_VALUE pour les événements suivis d'un EVENT_TYPE correspondent au pourcentage d'une vidéo regardée par un utilisateur, si vous définissez le seuil de valeur de l'événement à 0,5 et le type d'événement à regarder, Amazon Personalize entraîne le modèle en utilisant uniquement les événements d'interaction avec le watch dont la valeur EVENT_VALUE est supérieure ou égale à 0,5.

Les poids associés aux types d'événements détermineront leur importance. Un type d'événement plus important incitera le modèle entraîné à recommander plus probablement un élément susceptible d'interagir avec ce type d'événement. Par exemple, si vous avez défini « achat » avec un poids supérieur à celui de « clic » et que le modèle apprend qu'un utilisateur cliquera sur l'article B ou achètera l'article C grâce à l'historique de ses interactions, le modèle classera l'article C plus haut.

Pour optimiser une solution avec une configuration d'événements, vous créez une nouvelle solution avec la recette User-Personalization-v 2 ou la recette Personalized-Ranking-v 2 et spécifiez une configuration d'événements. Vous pouvez également mettre à jour une solution existante (Mettre à jour une solution pour modifier sa configuration d'entraînement automatique) avec une configuration d'événements.

Vous pouvez utiliser la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Pour plus d'informations sur l'utilisation de la console Amazon Personalize, consultezCréation d'une solution (console).

Rubriques

- Directives et exigences
- Mesurer les performances avec la configuration de la pondération des événements
- Optimisation d'une solution (AWS CLI)
- Optimisation d'une solution (AWS SDKs)

Directives et exigences

Les directives et les exigences relatives à la configuration des événements sont les suivantes :

- Pour configurer les pondérations pour différents types d'événements, votre jeu de données sur les interactions entre articles doit comporter une colonne EVENT_TYPE et éventuellement une colonne EVENT_VALUE.
- Vous pouvez spécifier une liste de paramètres d'événements dans la configuration. Incluez tous les types d'événements que vous souhaitez prendre en compte pour la création de la solution. Vous pouvez spécifier un maximum de 10 types d'événements différents.
- Vous pouvez spécifier le poids de l'événement pour chaque type d'événement. Le poids de l'événement doit être compris entre 0,0 et 1,0. Seul le rapport des poids entre les types d'événements est important. Par exemple, définir un type d'événement « achat » avec un poids de 0,3 et un type d'événement « clic » avec un poids de 0,1 aura le même effet que définir « acheter avec un poids de 0,6 » et un « clic » avec un poids de 0,2.
- Vous pouvez mettre à jour la configuration des événements pour une solution existante à l'aide de l'opération UpdateSolutionAPI.

Mesurer les performances avec la configuration de la pondération des événements

Lorsque vous créez une version de solution (entraînez un modèle) pour une solution avec une configuration d'événements, Amazon Personalize génère une `normalized_discounted_cumulative_gain_with_event_weights_at_k` métrique. Le score de `normalized_discounted_cumulative_gain_with_event_weights_at_k` indique les performances de la version de la solution compte tenu de la pondération des événements que vous avez définie pour chaque type d'événement.

Il est similaire au gain cumulé actualisé normalisé (NDCG) à K, mais la récompense pour chaque prédiction correcte sera pondérée. En revanche, dans le NDCG original à K, chaque prédiction correcte aura toutes une pondération de 1. Par exemple, avec un « achat » d'un poids de 0,3 et un « clic » d'un poids de 0,1, la prédiction correcte de l'article « d'achat » donnera droit à une récompense de 1,5 tandis que la prédiction d'un article « clic » obtiendra une récompense de 0,5.

Pour plus d'informations sur la génération de métriques, consultezÉvaluation d'une version de la solution Amazon Personalize à l'aide de métriques.

Optimisation d'une solution (AWS CLI)

Vous pouvez optimiser la configuration des événements avec la recette User-Personalization-v Personalized-Ranking-v 2 ou 2.

Pour optimiser une solution avec une configuration d'événements à l'aide du AWS CLI, créez une nouvelle solution et spécifiez les détails de configuration de vos événements à l'aide de la `eventsConfig` clé de l'`solutionConfig`objet. `eventsConfig`Il possède une clé `eventParametersList` sous laquelle vous pouvez spécifier jusqu'à 10 EventParameters. Chacun `eventParameter` comporte les champs suivants :

- EventType : spécifiez le type d'événement que vous souhaitez prendre en compte pour la création de la solution.
- eventValueThreshold: spécifiez le seuil de valeur de l'événement. Seuls les événements dont la valeur d'événement est supérieure ou égale à ce seuil seront pris en compte pour la création de la solution.
- poids : spécifiez le poids pour chaque type d'événement. Une pondération plus élevée signifie une plus grande importance du type d'événement pour la solution créée.

Voici un exemple de commande create-solution AWS CLI . Remplacez les `recipe arn` valeurs `solution namedataset group arn`, et par les vôtres.

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group arn \
--recipe-arn recipe arn \
--solution-config "{\"eventsConfig\":{\"eventParametersList\":[{\"eventType\":
\"Purchase\", \"eventValueThreshold\":0.1, \"weight\":0.3}, {\"eventType\":\"Click\",
 \"weight\":0.1}]}"
```

Lorsque votre solution est prête, créez une nouvelle version de solution (pour un exemple de commande, voirCréation d'une solution (AWS CLI)). Une fois que vous avez créé une version de solution, vous pouvez visualiser les performances d'optimisation à l'aide des métriques de version de solution. Consultez Mesurer les performances d'optimisation.

Optimisation d'une solution (AWS SDKs)

Vous pouvez optimiser la configuration des événements avec la recette User-Personalization-v Personalized-Ranking-v 2 ou 2.

Pour optimiser une solution avec une configuration d'événements à l'aide du AWS SDKs, créez une nouvelle solution et spécifiez les détails de configuration de vos événements à l'aide de la `eventsConfig` clé de l'`solutionConfig`objet. `eventsConfig`Il possède une clé `eventParametersList` sous laquelle vous pouvez spécifier jusqu'à 10`eventParameters`. Chacun `eventParameter` comporte les champs suivants :

- EventType : spécifiez le type d'événement que vous souhaitez prendre en compte pour la création de la solution.
- eventValueThreshold: spécifiez le seuil de valeur de l'événement. Seuls les événements dont la valeur d'événement est supérieure ou égale à ce seuil seront pris en compte pour la création de la solution.
- poids : spécifiez le poids pour chaque type d'événement. Une pondération plus élevée signifie une plus grande importance du type d'événement pour la solution créée.

SDK for Python (Boto3)

```
import boto3
```

```
personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name= 'solution name',
    recipeArn = 'recipe arn',
    datasetGroupArn = 'dataset group arn',
    solutionConfig = {
       "eventsConfig": {
          "eventParametersList": [
             {"eventType":"Purchase",
              "eventValueThreshold":0.1,
              "weight":0.3},
             {"eventType":"Click",
              "weight":0.1}
          ]
       }
    }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for Java 2.x

```
public static String createPersonalizeSolution(PersonalizeClient personalizeClient,
         String datasetGroupArn,
         String solutionName,
         String recipeArn,
         ) {

try {
    EventsConfig eventsConfig = EventsConfig.builder()
        .eventsParameterList(eventsParameterList)
        .build();

    SolutionConfig solutionConfig = SolutionConfig.builder()
        .eventsConfig(eventsConfig)
        .build();

    CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
        .name(solutionName)
        .datasetGroupArn(datasetGroupArn)
        .recipeArn(recipeArn)
        .solutionConfig(solutionConfig)
```

```
        .build();

    CreateSolutionResponse solutionResponse =
 personalizeClient.createSolution(solutionRequest);

    return solutionResponse.solutionArn();

} catch (PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
```

Lorsque votre solution est prête, créez une nouvelle version de solution (pour un exemple de commande, voirCréation d'une solution (AWS SDKs)). Une fois que vous avez créé une version de solution, vous pouvez visualiser les performances d'optimisation à l'aide des métriques de version de solution. Consultez Mesurer les performances d'optimisation.

Hyperparamètres et HPO

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Vous spécifiez les hyperparamètres avant l'entraînement afin d'optimiser le modèle entraîné pour votre cas d'utilisation particulier. Ils sont différents des paramètres de modèle, dont la valeur est déterminée au cours du processus de formation.

Les hyperparamètres sont spécifiés en utilisant la clé `algorithmHyperParameters` qui fait partie de l'objet SolutionConfig qui est transmis à l'opération CreateSolution.

Une version condensée de la `CreateSolution` demande se trouve ci-dessous. L'exemple inclut l'`solutionConfig`objet. Vous pouvez l'`solutionConfig`utiliser pour remplacer les paramètres par défaut d'une recette.

```
{
  "name": "string",
  "recipeArn": "string",
  "eventType": "string",
  "solutionConfig": {
      "optimizationObjective": {
          "itemAttribute": "string",
          "objectiveSensitivity": "string"
      },
      "eventValueThreshold": "string",
      "featureTransformationParameters": {
          "string" : "string"
      },
      "algorithmHyperParameters": {
          "string" : "string"
      },
      "hpoConfig": {
          "algorithmHyperParameterRanges": {
              ...
          },
          "hpoResourceConfig": {
              "maxNumberOfTrainingJobs": "string",
              "maxParallelTrainingJobs": "string"
          }
      },
  },
}
```

Différentes recettes utilisent différents hyperparamètres. Pour les hyperparamètres disponibles, consultez les recettes individuelles dans Choisir une recette.

Activation de l'optimisation des hyperparamètres

L'optimisation d'hyperparamètres (HPO, Hyperparameter Optimization) ou leur réglage consiste à choisir les hyperparamètres optimaux pour un objectif d'apprentissage spécifique. Les hyperparamètres optimaux sont déterminés en exécutant de nombreuses tâches de formation à l'aide des valeurs différentes depuis les plages de possibilités spécifiées.

Avec User-Personalization-v2et Personalized-Ranking-v2, si vous activez la formation automatique, Amazon Personalize effectue automatiquement le HPO tous les 90 jours. Sans entraînement automatique, aucun HPO ne se produit. Pour toutes les autres recettes, vous devez activer HPO. Pour utiliser HPO, définissez `performHP0` sur `true`, et incluez l'objet `hpoConfig`.

Les hyperparamètres peuvent être catégoriques, continus ou à valeurs entières. L'objet `hpoConfig` comporte les clés qui correspondent à chacun de ces types, où vous spécifiez les hyperparamètres et leurs plages. Vous devez fournir chaque type dans votre demande, mais si une recette n'a pas de paramètre de type, vous pouvez le laisser vide. Par exemple, User-Personalization n'a pas d'hyperparamètre réglable de type continu. Donc, pour le`continousHyperParameterRange`, vous devez transmettre un tableau vide.

Le code suivant montre comment créer une solution avec HPO activé à l'aide du AWS SDK pour Python (Boto3). La solution de l'exemple utilise la Recette de personnalisation par l'utilisateur recette et HPO est défini sur. `true` Le code fournit une valeur pour `hidden_dimension` et le `categoricalHyperParameterRanges` et`integerHyperParameterRanges`. Le `continousHyperParameterRange` est vide et `hpoResourceConfig` définit le `maxNumberOfTrainingJobs` et`maxParallelTrainingJobs`.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = "solution name",
    datasetGroupArn = 'arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName',
    recipeArn = 'arn:aws:personalize:::recipe/aws-user-personalization',
    performHPO = True,
    solutionConfig = {
        "algorithmHyperParameters": {
          "hidden_dimension": "55"
        },
        "hpoConfig": {
          "algorithmHyperParameterRanges": {
              "categoricalHyperParameterRanges": [
                  {
                      "name": "recency_mask",
                      "values": [ "true", "false"]
                  }
              ],
              "integerHyperParameterRanges": [
                  {
                      "name": "bptt",
                      "minValue": 2,
                      "maxValue": 22
                  }
```

```
            ],
            "continuousHyperParameterRanges": [

            ]
        },
        "hpoResourceConfig": {
            "maxNumberOfTrainingJobs": "4",
            "maxParallelTrainingJobs": "2"
        }
    }
}
)
```

Pour plus d'informations sur le HPO, consultez la section Réglage automatique du modèle.

Affichage des hyperparamètres

Vous pouvez afficher les hyperparamètres de la solution en appelant l'DescribeSolutionopération. L'exemple suivant montre une `DescribeSolution` sortie. Après avoir créé une version de solution (entraînement d'un modèle), vous pouvez également afficher les hyperparamètres associés à l'DescribeSolutionVersionopération.

```
{
  "solution": {
    "name": "hpo_coonfig_solution",
    "solutionArn": "arn:aws:personalize:region:accountId:solution/solutionName",
    "performHPO": true,
    "performAutoML": false,
    "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
    "datasetGroupArn": "arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName",
    "eventType": "click",
    "solutionConfig": {
      "hpoConfig": {
        "hpoResourceConfig": {
          "maxNumberOfTrainingJobs": "4",
          "maxParallelTrainingJobs": "2"
        },
        "algorithmHyperParameterRanges": {
          "integerHyperParameterRanges": [
            {
              "name": "training.bptt",
              "minValue": 2,
```

```
            "maxValue": 22
          }
        ],
        "continuousHyperParameterRanges": [],
        "categoricalHyperParameterRanges": [
          {
            "name": "data.recency_mask",
            "values": [
              "true",
              "false"
            ]
          }
        ]
      }
    },
    "algorithmHyperParameters": {
      "hidden_dimension": "55"
    }
  },
  "status": "ACTIVE",
  "creationDateTime": "2022-07-08T12:12:48.565000-07:00",
  "lastUpdatedDateTime": "2022-07-08T12:12:48.565000-07:00"
 }
}
```


Choix des données d'interaction entre les objets utilisées pour l'entraînement

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Vous pouvez choisir les événements dans un ensemble de données d'interactions entre articles qu'Amazon Personalize utilise lors de la création d'une version de solution (formation d'un modèle). Le choix des données d'interaction entre les éléments avant l'entraînement vous permet de n'utiliser

qu'un sous-ensemble pertinent de vos données pour l'entraînement ou de supprimer le bruit pour entraîner un modèle plus optimisé. Pour plus d'informations sur les ensembles de données d'interactions entre éléments, consultezDonnées d'interaction entre les articles.

> **Note**
>
> Si vous utilisez User-Personalization-v2 ou Personalized-Ranking-v2, vos frais de formation sont basés sur les données relatives aux interactions avec les objets avant d'être filtrés par type ou valeur d'événement. Pour plus d'informations sur la tarification, consultez la section Tarification Amazon Personalize.

Vous pouvez sélectionner les données d'interaction entre les articles comme suit :

- Choisissez les enregistrements en fonction du type : lorsque vous configurez une solution, si votre jeu de données d'interactions avec les articles inclut des types d'événements dans une colonne EVENT_TYPE, vous pouvez éventuellement spécifier un type d'événement à utiliser dans le cadre de l'entraînement. Par exemple, si votre ensemble de données d'interactions avec les articles inclut des types d'événements d'achat, de clic et de surveillance, et que vous souhaitez qu'Amazon Personalize entraîne le modèle avec uniquement des événements de surveillance, lorsque vous configurez votre solution, vous devez fournir une surveillance sous la forme `event type` que Amazon Personalize utilise dans le cadre de la formation.

  Si vous avez plusieurs types d'événements et que vous utilisez la User-Personalization-v2 recette ou la Personalized-Ranking-v2 recette, lorsque vous configurez une solution personnalisée, vous pouvez spécifier des pondérations différentes pour les différents types. Par exemple, vous pouvez configurer une solution pour donner plus de poids aux événements d'achat qu'aux événements de clics. Pour plus d’informations, consultez Optimisation d'une solution avec configuration d'événements.

  Si votre ensemble de données d'interactions avec les articles contient plusieurs types d'événements dans une colonne EVENT_TYPE, et que vous ne fournissez aucun type d'événement lorsque vous configurez votre solution, Amazon Personalize utilise toutes les données d'interaction avec les articles pour l'entraînement avec le même poids, quel que soit le type.
- Choisissez les enregistrements en fonction du type et de la valeur : lorsque vous configurez une solution, si votre jeu de données d'interactions avec les articles inclut les champs EVENT_TYPE et EVENT_VALUE, vous pouvez définir une valeur spécifique comme seuil pour exclure les enregistrements de l'entraînement. Par exemple, si vos données EVENT_VALUE pour les

événements suivis d'un EVENT_TYPE correspondent au pourcentage d'une vidéo regardée par un utilisateur, si vous définissez le seuil de valeur de l'événement à 0,5 et le type d'événement à regarder, Amazon Personalize entraîne le modèle en utilisant uniquement les événements d'interaction avec le watch dont la valeur EVENT_VALUE est supérieure ou égale à 0,5.

Le code suivant montre comment utiliser le AWS SDK pour Python (Boto3) pour créer une solution qui utilise uniquement les `watch` événements pour lesquels l'utilisateur a visionné plus de la moitié de la vidéo.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = 'solution name',
    datasetGroupArn = 'arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName',
    recipeArn = 'arn:aws:personalize:::recipe/aws-user-personalization-v2',
    eventType = 'watch',
    solutionConfig = {
        "eventValueThreshold": "0.5"
    }
)

# Store the solution ARN
solution_arn = create_solution_response['solutionArn']

# Use the solution ARN to get the solution status
solution_description = personalize.describe_solution(solutionArn = solution_arn)
['solution']
print('Solution status: ' + solution_description['status'])
```

Clonage d'une solution (console)

Lorsque vous créez une nouvelle solution, vous pouvez utiliser la console Amazon Personalize pour cloner une solution. Lorsque vous clonez une solution, vous pouvez utiliser la configuration de la solution existante comme point de départ, comme la recette et les hyperparamètres, et apporter les modifications nécessaires. Cela est utile si vous souhaitez apporter une modification à une solution sans modifier toutes les autres propriétés. Par exemple, ajouter une nouvelle colonne de données d'entraînement à votre ensemble de données. Dans ce cas, vous devez cloner une solution, lui

donner un nom, modifier les colonnes utilisées lors de l'entraînement et laisser toutes les autres propriétés inchangées.

Cloner une solution

Pour cloner une solution, vous choisissez la solution existante, puis l'option Cloner la solution. Donnez ensuite un nom à la nouvelle solution et modifiez les champs correspondants.

Pour cloner une solution

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Choisissez Ressources personnalisées, puis Solutions.
4. Choisissez la solution que vous souhaitez cloner.
5. Choisissez Actions, puis choisissez la solution Cloner.
6. Donnez un nom à la nouvelle solution.
7. Apportez des modifications aux détails de la solution et à la configuration avancée. Amazon Personalize préremplit ces champs avec des valeurs issues de la solution existante. Pour plus d'informations sur chaque champ, consultezConfiguration d'une solution personnalisée dans Amazon Personalize.

Mettre à jour une solution pour modifier sa configuration d'entraînement automatique

Après avoir créé une solution, vous pouvez modifier sa configuration d'entraînement automatique et sa configuration d'événements :

- Vous pouvez activer ou désactiver l'entraînement automatique et modifier la fréquence d'entraînement.
  - Si vous activez l'entraînement automatique, le premier entraînement automatique démarre dans l'heure qui suit la fin de la mise à jour de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore le premier entraînement automatique.
  - Si vous modifiez la fréquence d'entraînement de la solution, le programme de formation est réinitialisé et une nouvelle version de la solution démarre l'entraînement dans l'heure qui suit. La

création de la version de la solution se poursuit à la nouvelle fréquence, le premier jour étant le jour de mise à jour de la solution.

- Vous pouvez mettre à jour la configuration des événements des solutions. Si une configuration d'événements est déjà spécifiée dans la solution, la nouvelle configuration d'événements remplacera la configuration d'origine. Pour plus d'informations sur l'optimisation d'une solution pour la configuration d'un événement, consultezOptimisation d'une solution avec configuration d'événements.

Vous pouvez mettre à jour une solution à l'aide de la console Amazon Personalize ou AWS des kits SDK. AWS Command Line Interface Les mises à jour de la solution peuvent prendre quelques minutes. Pendant que la mise à jour est en cours, vous pouvez créer des versions de solution pour la solution, mais vous ne pouvez pas supprimer la solution. Jusqu'à ce que la mise à jour soit terminée, la solution utilise la configuration précédente. Pour plus d'informations sur l'entraînement automatique, consultezConfiguration de l'entraînement automatique.

Rubriques

- Mettre à jour une solution (console)
- Mettre à jour une solution (AWS CLI)
- Mettre à jour une solution (AWS SDK)

Mettre à jour une solution (console)

Pour mettre à jour une solution dans la console, accédez à la solution, choisissez Mettre à jour et spécifiez la nouvelle configuration à utiliser.

Pour configurer une solution

1. Ouvrez la console Amazon Personalize à https://console.aws.amazon.com/personalize/homel'adresse et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sélectionnez Ressources personnalisées, puis Solutions et recettes.
4. Choisissez votre solution et choisissez Mettre à jour en haut à droite.
5. Dans Entraînement automatique, indiquez si la solution utilise l'entraînement automatique. Si l'entraînement automatique est activé, vous pouvez modifier le`Automatic training frequency`. La fréquence d'entraînement par défaut est tous les 7 jours.

6. Choisissez Mettre à jour la solution. Vous trouverez l'état de la mise à jour de la solution sur la page de détails de votre solution.

Mettre à jour une solution (AWS CLI)

Pour mettre à jour une solution avec le AWS Command Line Interface, utilisez la `update-solution` commande. Cette commande utilise l'opération UpdateSolution API. Le code suivant explique comment mettre à jour une solution pour utiliser l'entraînement automatique avec une fréquence d'entraînement de 5 jours. Pour désactiver l'entraînement automatique, spécifiez `--no-perform-auto-training` et omettez le`solution-update-config`.

La fréquence d'entraînement par défaut est tous les 7 jours. L'expression doit être au `rate(`*`value unit`*`)` format. Pour la valeur, spécifiez un nombre compris entre 1 et 30. Pour l'unité, spécifiez `day` ou`days`.

```
aws personalize update-solution \
--solution-arn solution ARN \
--perform-auto-training \
--solution-update-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5
 days)\"}}"
```

Pour connaître l'état de la mise à jour, utilisez la `describe-solution` commande (qui utilise l'opération DescribeSolution API) et recherchez l'état de la mise à jour dans le`latestSolutionUpdate`.

Mettre à jour une solution (AWS SDK)

Pour mettre à jour une solution avec AWS les SDK, utilisez l'opération UpdateSolution API. Le code suivant vous montre comment utiliser le pour mettre AWS SDK pour Python (Boto3) à jour une solution afin d'utiliser l'entraînement automatique avec une fréquence d'entraînement de 5 jours. Le code obtient le statut de la mise à jour grâce à l'opération DescribeSolution API.

La fréquence d'entraînement par défaut est tous les 7 jours. L'expression doit être au `rate(`*`value unit`*`)` format. Pour la valeur, spécifiez un nombre compris entre 1 et 30. Pour l'unité, spécifiez `day` ou`days`.

```
import boto3

personalize = boto3.client('personalize')
```

```
update_solution_response = personalize.update_solution(
    solutionArn='SOLUTION ARN',
    performAutoTraining=True,
    solutionUpdateConfig={
        "autoTrainingConfig": {
            "schedulingExpression": "rate(5 days)"
        }
    }
)
describe_solution_response = personalize.describe_solution(
    solutionArn='SOLUTION ARN'
)
update_status = describe_solution_response["solution"]["latestSolutionUpdate"]
["status"]
print(f"Update status: {update_status}")
```

Création manuelle d'une version de solution

Une fois que vous avez terminéConfiguration d'une solution personnalisée dans Amazon Personalize, vous êtes prêt à commencer l'entraînement :

- Si votre solution utilise l'entraînement automatique, elle crée des versions de solution pour vous à la fréquence d'entraînement que vous spécifiez. Par défaut, toutes les nouvelles solutions utilisent une formation automatique pour créer une nouvelle version de solution tous les 7 jours. Vous pouvez toujours créer des versions de solution manuellement. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique.
- Si vous désactivez l'entraînement automatique pour votre solution ou si vous souhaitez effectuer un entraînement manuel, vous pouvez créer manuellement une version de la solution. Une version de solution fait référence à un modèle d'apprentissage automatique entraîné. Vous pouvez créer une version de solution à l'aide de la console, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez utiliser cette the section called “StopSolutionVersionCreation” opération pour arrêter le processus de création de la version de solution. Consultez Arrêt de la création d'une version de solution.

Si la formation ne se termine pas en raison d'une erreur, elle ne vous sera pas facturée. Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez arrêter le processus de création de version de solution. Pour arrêter la création d'une version de

solution, accédez à la page de détails de la version de solution et choisissez Arrêter. Pour de plus amples informations, veuillez consulter Arrêt de la création d'une version de solution.

Rubriques

- Création d'une version de solution (console)
- Création d'une version de solution (AWS CLI)
- Création d'une version de solution (AWS SDKs)

Création d'une version de solution (console)

Pour créer manuellement une nouvelle version de solution à l'aide de la console Amazon Personalize, vous devez commencer la formation depuis la page de détails de votre solution.

Pour créer une nouvelle version de solution

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Accédez à la page des groupes de jeux de données et choisissez le groupe de jeux de données avec votre nouvelle solution.
3. Dans le volet de navigation, sous Ressources personnalisées, sélectionnez Solutions et recettes.
4. Sur la page Solution et recettes, choisissez la solution pour laquelle vous souhaitez créer une version de solution.
5. Sur la page de présentation de la solution, choisissez Créer une version de solution pour commencer à entraîner un nouveau modèle.

Sur la page des détails de la solution, vous pouvez suivre la progression de la formation dans la section Versions de la solution. Lorsque la formation est terminée, le statut est Actif, vous pouvez l'évaluer à l'aide des indicateurs fournis par Amazon Personalize. Pour de plus amples informations, veuillez consulter Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques.

Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

- Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des

recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.

- Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Création d'une version de solution (AWS CLI)

Lorsque votre solution est ACTIVE, entraînez le modèle en exécutant la commande suivante. `solution arn`Remplacez-le par la solution Amazon Resource Name (ARN) deConfiguration d'une solution personnalisée dans Amazon Personalize.

```
aws personalize create-solution-version \
  --solution-arn solution arn
```

L'ARN de version de solution est affiché, par exemple :

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>"
}
```

Vérifiez l'état d'entraînement de la version de la solution à l'aide de la `describe-solution-version` commande. Fournissez l'ARN de la version de solution qui a été renvoyé lors de l'étape précédente. Pour plus d'informations sur l'API, consultez DescribeSolutionVersion.

```
aws personalize describe-solution-version \
  --solution-version-arn solution version arn
```

Les propriétés de la version de solution et le `status` de formation sont affichés. Initialement, le statut indique CREATE PENDING (CRÉATION EN ATTENTE), par exemple :

```
{
  "solutionVersion": {
      "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
solutionName/<version-id>",
      ...,
      "status": "CREATE PENDING"
```

```
  }
}
```

La formation est terminée lorsqu'elle `status` est terminée `ACTIVE` et vous pouvez l'évaluer à l'aide des indicateurs fournis par Amazon Personalize. Pour de plus amples informations, veuillez consulter Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques. Si la formation ne se termine pas en raison d'une erreur, elle ne vous sera pas facturée.

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez utiliser cette StopSolutionVersionCreation opération pour arrêter le processus de création de la version de solution. Consultez Arrêt de la création d'une version de solution.

Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

- Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.
- Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Création d'une version de solution (AWS SDKs)

Lorsque votre solution est ACTIVE, utilisez le code suivant pour créer une version de solution. Spécifiez le nom de ressource Amazon (ARN) à partir deConfiguration d'une solution personnalisée dans Amazon Personalize. Utilisez l'DescribeSolutionVersionopération pour récupérer l'état de la version de la solution.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')
# Store the solution ARN
solution_arn = 'solution arn'
```

```
# Use the solution ARN to get the solution status.
solution_description = personalize.describe_solution(solutionArn = 'solution_arn')
['solution']
print('Solution status: ' + solution_description['status'])

# Use the solution ARN to create a solution version.
print ('Creating solution version')
response = personalize.create_solution_version(solutionArn = solution_arn)
solution_version_arn = response['solutionVersionArn']
print('Solution version ARN: ' + solution_version_arn)

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeSolutionVersion(PersonalizeClient
 personalizeClient, String solutionArn) {
        long maxTime = 0;
        long waitInMilliseconds = 30 * 1000; // 30 seconds
        String solutionStatus = "";
        String solutionVersionStatus = "";
        String solutionVersionArn = "";

        try {
            DescribeSolutionRequest describeSolutionRequest =
 DescribeSolutionRequest.builder()
                .solutionArn(solutionArn)
                .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            // Wait until solution is active.
            while (Instant.now().getEpochSecond() < maxTime) {

                solutionStatus =
 personalizeClient.describeSolution(describeSolutionRequest).solution().status();
                System.out.println("Solution status: " + solutionStatus);

                if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
 FAILED")) {
```

```
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }

            // Once the solution is active, start creating a solution version.

            if (solutionStatus.equals("ACTIVE")) {

                CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
                    .solutionArn(solutionArn)
                    .build();

                CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient.createSolutionVersion(createSolutionVersionRequest);
                solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

                System.out.println("Solution version ARN: " + solutionVersionArn);

                DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
                    .solutionVersionArn(solutionVersionArn)
                    .build();

                maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

                while (Instant.now().getEpochSecond() < maxTime) {

                    // Use the solution version ARN to get the solution version
status.
                    solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest).solutionVersion()
                    System.out.println("Solution version status: " +
solutionVersionStatus);

                    if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                        break;
```

```
                    }
                    try {
                        Thread.sleep(waitInMilliseconds);
                    } catch (InterruptedException e) {
                        System.out.println(e.getMessage());
                    }
                }
                return solutionVersionArn;
            }
        } catch(PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSolutionVersionCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution version parameters.
export const solutionVersionParam = {
  solutionArn: "SOLUTION_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionVersionCommand(solutionVersionParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Pour vérifier le statut de la version de solution actuelle, appelez l'opération DescribeSolutionVersion et transmettez l'ARN de la version de solution renvoyée à partir de l'opération `CreateSolutionVersion`. La formation est terminée lorsqu'elle `status` est terminée `ACTIVE` et vous pouvez l'évaluer à l'aide des indicateurs fournis par Amazon Personalize. Pour de plus amples informations, veuillez consulter Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques. Si la formation ne se termine pas en raison d'une erreur, elle ne vous sera pas facturée.

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez utiliser cette StopSolutionVersionCreation opération pour arrêter le processus de création de la version de solution. Consultez Arrêt de la création d'une version de solution.

Lorsque la version de la solution est ACTIVE, vous êtes prêt à l'utiliser pour obtenir des recommandations. La façon dont vous utilisez une version de solution active dépend de la manière dont vous obtenez les recommandations :

- Pour obtenir des recommandations en temps réel, vous déployez une version de la solution ACTIVE avec une campagne Amazon Personalize. Vous utilisez la campagne pour obtenir des recommandations pour vos utilisateurs. Consultez Déploiement d'une version de la solution Amazon Personalize avec une campagne.
- Pour les recommandations par lots, vous devez spécifier une version de solution ACTIVE lorsque vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Voir Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Arrêt de la création d'une version de solution

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez utiliser la console Amazon Personalize ou l'StopSolutionVersionCreationopération pour arrêter de créer la version de la solution (arrêter de former un modèle). Vous ne pouvez pas reprendre la création d'une version de solution une fois celle-ci arrêtée. Les ressources utilisées jusqu'à l'arrêt de la création de la version de la solution vous sont facturées.

L'arrêt de la création d'une version de solution met fin à l'entraînement du modèle, mais ne supprime pas la version de solution. Vous pouvez toujours consulter les détails de la version de la solution dans la console Amazon Personalize et lors de l'DescribeSolutionVersionopération.

Vous pouvez arrêter le processus de création de version de solution à l'aide de la console Amazon Personalize, du AWS Command Line Interface (AWS CLI) ou du AWS SDKs.

Rubriques

- Arrêt de la création d'une version de solution (console)
- Arrêt de la création d'une version de solution (AWS CLI)
- Arrêt de la création d'une version de solution (AWS SDKs)

Arrêt de la création d'une version de solution (console)

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez arrêter de créer une version de solution (arrêter de former un modèle).

Pour arrêter de créer une version de solution (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez le groupe de jeux de données contenant la version de solution que vous souhaitez arrêter.
3. Dans le volet de navigation, sélectionnez Solutions et recettes.
4. Sur la page Solution et recettes, choisissez la solution avec la version de solution que vous souhaitez arrêter.
5. Dans Versions de solution, choisissez la version de solution que vous souhaitez arrêter.
6. Sur la page des détails de la version de la solution, choisissez Arrêter la création. En fonction de l'état d'origine de la version de solution, l'état de la version de solution change comme suit :
   - CREATE_PENDING devient CREATE_STOPPED.
   - CREATE_IN_PROGRESS devient CREATE_STOPPING puis CREATE_STOPPED.

Arrêt de la création d'une version de solution (AWS CLI)

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez arrêter de créer une version de solution (arrêter de former un modèle). Utilisez la `stop-solution-version-creation` commande suivante pour arrêter de créer la version de la solution avec AWS CLI. `solution version arn`Remplacez-le par le Amazon Resource Name (ARN) de la version de solution que vous souhaitez arrêter. Les ressources utilisées jusqu'à l'arrêt de la création de la version de la solution vous sont facturées.

```
aws personalize stop-solution-version-creation \
```

```
    --solution-version-arn solution version arn
```

Vérifiez l'état d'entraînement de la version de la solution à l'aide de la `describe-solution-version` commande.

```
aws personalize describe-solution-version \
    --solution-version-arn solution version arn
```

En fonction de l'état d'origine de la version de solution, l'état de la version de solution change comme suit :

- CREATE_PENDING devient CREATE_STOPPED.
- CREATE_IN_PROGRESS devient CREATE_STOPPING puis CREATE_STOPPED

Arrêt de la création d'une version de solution (AWS SDKs)

Si le statut de la version de votre solution est CREATE_PENDING ou CREATE_IN_PROGRESS, vous pouvez arrêter de créer une version de solution (arrêter de former un modèle). Le code suivant montre comment arrêter de créer une version de solution avec le AWS SDK pour Python (Boto3) ou AWS SDK for Java 2.x. Les ressources utilisées jusqu'à l'arrêt de la création de la version de la solution vous sont facturées.

SDK for Python (Boto3)

Utilisez la `stop_solution_version_creation` méthode suivante pour arrêter la création d'une version de solution. `solution_version_arn`Remplacez-le par le Amazon Resource Name (ARN) de la version de solution que vous souhaitez arrêter. La méthode utilise l'DescribeSolutionVersionopération pour récupérer l'état de la version de la solution.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.stop_solution_version_creation(
    solutionVersionArn = solution_version_arn
)

# Use the solution version ARN to get the solution version status.
```

```
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

Utilisez la `stopSolutionVersionCreation` méthode suivante pour arrêter de créer une version de solution. Passez en paramètres un client du service Amazon Personalize et le nom de ressource Amazon (ARN) de la version de solution que vous souhaitez arrêter de créer. Le code suivant utilise l'DescribeSolutionVersionopération pour récupérer l'état de la version de la solution.

```
public static void stopSolutionVersionCreation(PersonalizeClient personalizeClient,
 String solutionVersionArn) {
    String solutionVersionStatus = "";

    StopSolutionVersionCreationRequest stopSolutionVersionCreationRequest =
 StopSolutionVersionCreationRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .build();


 personalizeClient.stopSolutionVersionCreation(stopSolutionVersionCreationRequest);

    // Use the solution version ARN to get the solution version status.
    DescribeSolutionVersionRequest describeSolutionVersionRequest =
 DescribeSolutionVersionRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .build();

    solutionVersionStatus =
 personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
        .solutionVersion()
        .status();
    System.out.println("Solution version status: " + solutionVersionStatus);
}
```

En fonction de l'état d'origine de la version de solution, l'état de la version de solution change comme suit :

- CREATE_PENDING devient CREATE_STOPPED.

- CREATE_IN_PROGRESS devient CREATE_STOPPING puis CREATE_STOPPED.

Évaluation d'une version de la solution Amazon Personalize à l'aide de métriques

Vous pouvez évaluer les performances de la version de votre solution grâce à des indicateurs en ligne et hors ligne. Les métriques en ligne sont les résultats empiriques que vous observez dans les interactions de vos utilisateurs avec des recommandations en temps réel. Par exemple, vous pouvez enregistrer le taux de clics de vos utilisateurs lorsqu'ils parcourent votre catalogue. Vous êtes responsable de la génération et de l'enregistrement de tous les indicateurs en ligne.

Les métriques hors ligne sont les métriques générées par Amazon Personalize lorsque vous entraînez une version de solution. Les métriques hors ligne vous permettent d'évaluer les performances du modèle. Vous pouvez visualiser les effets de la modification des hyperparamètres d'une solution et comparer les résultats de modèles entraînés avec différentes recettes sur les mêmes données dans le même groupe de jeux de données.

Évitez de comparer les métriques des différentes versions de solutions entraînées avec différentes données. La différence entre les métriques peut être due à la différence entre les données plutôt qu'aux performances du modèle. Par exemple, vous pouvez avoir un groupe de jeux de données contenant des données d'`purchase`événements éparses pour chaque utilisateur, et un autre contenant des données d'`view`événements robustes. Sur la base de mesures telles que `precision at K` la version de la solution entraînée sur les données d'événements de visualisation peut sembler à tort plus performante en raison du nombre plus élevé d'interactions.

Pour obtenir des indicateurs de performance, Amazon Personalize divise les données des interactions d'entrée en un ensemble d'entraînement, un ensemble de test et, pour PERSONALIZED_ACTIONS, un ensemble de validation. La répartition dépend du type de recette que vous choisissez :

- Pour les recettes USER_SEGMENTATION, le kit de formation comprend 80 % des données d'interactions de chaque utilisateur et le kit de test comprend 20 % des données d'interactions de chaque utilisateur.
- Pour tous les autres types de recettes, le kit d'entraînement comprend 90 % de vos utilisateurs et de leurs données d'interaction. L'ensemble de tests comprend les 10 % d'utilisateurs restants et leurs données d'interactions.

Amazon Personalize crée ensuite la version de la solution à l'aide du kit de formation. Une fois la formation terminée, Amazon Personalize fournit à la nouvelle version de la solution les 90 % les plus anciens des données de chaque utilisateur issues du kit de test en entrée. Amazon Personalize calcule ensuite les indicateurs en comparant les recommandations générées par la version de la solution aux interactions réelles figurant dans les 10 % les plus récents des données de chaque utilisateur issues du kit de test.

Pour générer une base de référence à des fins de comparaison, nous vous recommandons d'utiliser la Comptage de popularité recette, qui recommande les K articles les plus populaires.

Rubriques

- Récupération des métriques de version de la solution
- Définitions des métriques
- Exemple
- Ressources supplémentaires

Récupération des métriques de version de la solution

Après avoir créé une version de solution, vous pouvez utiliser des métriques pour évaluer ses performances. Vous pouvez récupérer les métriques d'une version de solution à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) et AWSSDKs.

Rubriques

- Récupération des métriques de version de la solution (console)
- Récupération des métriques de version de la solution () AWS CLI
- Récupération des métriques de version de la solution () AWS SDKs

Récupération des métriques de version de la solution (console)

Pour consulter les statistiques des recommandations dans la console, accédez à la page de détails de la version de votre solution.

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données personnalisé.

3. Dans le volet de navigation, sélectionnez Ressources personnalisées, puis Solutions et recettes.
4. Choisissez votre solution.
5. Dans Versions de solution, choisissez la version de votre solution pour afficher sa page de détails. Les métriques sont répertoriées dans l'onglet Mesures de version de solution dans le volet inférieur. Pour les définitions des métriques, voirDéfinitions des métriques.

   Maintenant que vous avez évalué la version de votre solution, vous pouvez créer une campagne en déployant la version de la solution présentant les meilleurs indicateurs pour votre cas d'utilisation. Pour plus d'informations sur le déploiement d'une solution, consultezDéploiement d'une version de la solution Amazon Personalize avec une campagne.

Récupération des métriques de version de la solution () AWS CLI

Vous pouvez extraire les métriques pour une version de solutions spécifique en appelant l'opération GetSolutionMetrics. Le code suivant montre comment récupérer des métriques à l'aide duAWS CLI.

```
personalize get-solution-metrics --solution-version-arn solution version ARN
```

Voici un exemple de sortie d'une version de solution créée à l'aide de la Personnalisation par l'utilisateur recette avec un objectif d'optimisation supplémentaire.

```
{
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>",
    "metrics": {
        "coverage": 0.27,
        "mean_reciprocal_rank_at_25": 0.0379,
        "normalized_discounted_cumulative_gain_at_5": 0.0405,
        "normalized_discounted_cumulative_gain_at_10": 0.0513,
        "normalized_discounted_cumulative_gain_at_25": 0.0828,
        "precision_at_5": 0.0136,
        "precision_at_10": 0.0102,
        "precision_at_25": 0.0091,
        "average_rewards_at_k": 0.653
    }
}
```

Pour obtenir des explications sur chaque métrique, voirDéfinitions des métriques. Maintenant que vous avez évalué la version de votre solution, vous pouvez créer une campagne en déployant la

version de la solution présentant les meilleurs indicateurs pour votre cas d'utilisation. Pour plus d'informations sur le déploiement d'une solution, consultezDéploiement d'une version de la solution Amazon Personalize avec une campagne.

Récupération des métriques de version de la solution () AWS SDKs

Vous pouvez extraire les métriques pour une version de solutions spécifique en appelant l'opération GetSolutionMetrics. Utilisez le code suivant pour récupérer les mesures.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.get_solution_metrics(
    solutionVersionArn = 'solution version arn')

print(response['metrics'])
```

SDK for Java 2.x

```
public static void getSolutionVersionMetrics(PersonalizeClient personalizeClient,
 String solutionVersionArn) {

    try {
        GetSolutionMetricsRequest request = GetSolutionMetricsRequest.builder()
                .solutionVersionArn(solutionVersionArn)
                .build();
        Map<String, Double> metrics =
 personalizeClient.getSolutionMetrics(request).metrics();
        metrics.forEach((key, value) -> System.out.println(key + " " + value));
    } catch (PersonalizeException e ) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Voici un exemple de sortie d'une version de solution créée à l'aide de la Personnalisation par l'utilisateur recette avec un objectif d'optimisation supplémentaire.

```
{
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
    "metrics": {
        "coverage": 0.27,
        "mean_reciprocal_rank_at_25": 0.0379,
        "normalized_discounted_cumulative_gain_at_5": 0.0405,
        "normalized_discounted_cumulative_gain_at_10": 0.0513,
        "normalized_discounted_cumulative_gain_at_25": 0.0828,
        "precision_at_5": 0.0136,
        "precision_at_10": 0.0102,
        "precision_at_25": 0.0091,
        "average_rewards_at_k": 0.653
    }
}
```

Pour obtenir des explications sur chaque métrique, voirDéfinitions des métriques. Maintenant que vous avez évalué la version de votre solution, vous pouvez créer une campagne en déployant la version de la solution présentant les meilleurs indicateurs pour votre cas d'utilisation. Pour plus d'informations sur le déploiement d'une solution, consultezDéploiement d'une version de la solution Amazon Personalize avec une campagne.

Définitions des métriques

Les métriques générées par Amazon Personalize pour les versions des solutions sont décrites ci-dessous en utilisant les termes suivants :

- Une recommandation pertinente est une recommandation pour un élément avec lequel l'utilisateur a réellement interagi. Ces éléments proviennent des 10 % les plus récents des données d'interactions de chaque utilisateur issues du kit de test.
- Rang fait référence à la position d'un élément dans la liste des recommandations. La position 1 (en haut de la liste) est présumée comme étant la plus pertinente pour l'utilisateur.

Pour chaque métrique, les valeurs les plus élevées (plus proches de 1) sont préférables. Pour en savoir plus, consultez les ressources répertoriées dansRessources supplémentaires.

couverture

La valeur de couverture vous indique la proportion d'articles uniques (pour les recommandations d'articles), d'actions (pour les recommandations d'actions) ou d'utilisateurs (pour les

recommandations de segments d'utilisateurs) qu'Amazon Personalize peut recommander par rapport au nombre total d'enregistrements uniques de vos ensembles de données.

Un score de couverture plus élevé signifie qu'Amazon Personalize recommande une plus grande partie de votre catalogue, plutôt que les mêmes enregistrements à plusieurs reprises. Les recettes qui proposent l'exploration d'objets, telles que la personnalisation par l'utilisateur, ont une couverture plus étendue que celles qui n'en proposent pas, telles que les recettes similaires.

rang réciproque moyen à 25

Cette métrique indique la capacité d'un modèle à générer des recommandations d'articles pertinentes à la première place.

Vous pouvez choisir un modèle dont le classement réciproque moyen est élevé (25) si vous générez des résultats de recherche d'articles pour un utilisateur et ne vous attendez pas à ce que l'utilisateur choisisse un article plus bas dans la liste. Par exemple, les utilisateurs choisissent fréquemment la première recette de cuisine dans les résultats de recherche. Amazon Personalize ne génère pas cette métrique pour les recettes PERSONALIZED_ACTIONS ou USER_SEGMENTATION.

Amazon Personalize calcule cette métrique en utilisant le score de classement réciproque moyen pour les demandes de recommandations. Chaque score de classement réciproque est calculé comme suit :`1 / the rank of the highest item interacted with by the user`, où le total des classements possibles est de 25. Les autres éléments de rang inférieur avec lesquels l'utilisateur interagit sont ignorés. Si l'utilisateur a choisi le premier élément, le score est de 1. S'ils ne choisissent aucun objet, le score est de 0.

Par exemple, vous pouvez présenter 25 recommandations à trois utilisateurs différents :

- Si l'utilisateur 1 clique sur l'objet au rang 4 et sur l'élément au rang 10, son score de classement réciproque est de 1/4.
- Si l'utilisateur 2 clique sur un élément au rang 2, un élément au rang 4 et un élément au rang 12, son score de classement réciproque est de 1/2.
- Si l'utilisateur 3 clique sur un seul élément au rang 6, son score de classement réciproque est de 1/6.

Le classement réciproque moyen de toutes les demandes de recommandations (dans ce cas 3) est calculé comme suit`(1/4 + 1/2 + 1/6) / 3 = .3056`.

gain cumulé actualisé normalisé (NDCG) à K (5/10/25)

Cette métrique indique dans quelle mesure votre modèle classe les recommandations d'éléments ou d'actions, où K est un échantillon de 5, 10 ou 25 recommandations. Cette métrique est utile si vous êtes particulièrement intéressé par le classement des recommandations au-delà de l'élément ou de l'action le mieux classé (pour cela, voir`mean reciprocal rank at 25`). Par exemple, le score de `NDCG at 10` serait utile si vous avez une application qui affiche jusqu'à 10 films dans un carrousel à la fois.

Amazon Personalize calcule le NDCG en attribuant une pondération aux recommandations en fonction de leur position dans le classement de chaque utilisateur du kit de test. Chaque recommandation est réduite (avec une pondération plus faible) selon un facteur qui dépend de sa position. La métrique finale est la moyenne `NDCG at K` pour tous les utilisateurs du kit de test. Cela `NDCG at K` suppose que les recommandations situées en bas d'une liste sont moins pertinentes que les recommandations situées en haut de la liste.

Amazon Personalize utilise un facteur de pondération de`1/log(1 + position)`, où le haut de la liste est la position`1`.

précision à K

Cette métrique indique dans quelle mesure les recommandations de votre modèle sont pertinentes sur la base d'un échantillon de K (5, 10 ou 25) recommandations.

Amazon Personalize calcule cette métrique en fonction du nombre de recommandations pertinentes parmi les K meilleures recommandations pour chaque utilisateur du kit de test, divisé par K, où K est 5, 10 ou 25. La mesure finale est la moyenne de tous les utilisateurs du kit de test.

Par exemple, si vous recommandez 10 éléments à un utilisateur et que l'utilisateur interagit avec 3 d'entre eux, la précision à K est de 3 éléments correctement prédits divisés par le total des 10 éléments recommandés :`3 / 10 = .30`.

Cette métrique récompense la recommandation précise d'articles pertinents. Plus le score est proche de 1, plus le modèle est précis.

precision

Si vous entraînez une version de solution avec la Next-Best-Action recette, Amazon Personalize génère une `precision` métrique au lieu de`precision at K`. Cette métrique vous indique dans quelle mesure votre modèle prédit les actions que les utilisateurs vont réellement entreprendre.

Pour calculer`precision`, pour chaque action de votre ensemble de données, Amazon Personalize divise le nombre d'utilisateurs correctement prévus pour effectuer l'action par le nombre total de fois que l'action a été recommandée. Amazon Personalize calcule ensuite la moyenne de toutes les actions de votre ensemble de données.

Par exemple, si une action a été recommandée à 100 utilisateurs, et que 60 utilisateurs l'ont effectuée et 40 utilisateurs qui ne l'ont pas fait, `precision` l'action est la suivante : `60 / 100 = .60` Amazon Personalize applique ensuite ce calcul à toutes les actions et renvoie la moyenne.

Cette métrique récompense la recommandation précise d'actions pertinentes. Plus le score est proche de 1, plus le modèle est précis.

récompense_moyen_at_k

Lorsque vous créez une version de solution (entraînez un modèle) pour une solution avec un objectif d'optimisation, Amazon Personalize génère une `average_rewards_at_k` métrique. Le score de `average_rewards_at_k` indique dans quelle mesure la version de la solution permet d'atteindre votre objectif. Pour calculer cette statistique, Amazon Personalize calcule les récompenses pour chaque utilisateur comme suit :

```
rewards_per_user = total rewards from the user's interactions with their
top 25 reward generating recommendations / total rewards from the user's
interactions with recommendations
```

La valeur finale `average_rewards_at_k` est la moyenne de toutes les valeurs `rewards_per_user` normalisées pour être une valeur décimale inférieure ou égale à 1 et supérieure à 0. Plus la valeur est proche de 1, plus vous pouvez vous attendre à des gains en moyenne par utilisateur grâce aux recommandations.

Par exemple, si votre objectif est de maximiser les revenus générés par les clics, Amazon Personalize calcule le score de chaque utilisateur en divisant le chiffre d'affaires total généré par les articles sur lesquels l'utilisateur a cliqué parmi ses 25 recommandations les plus chères par les revenus provenant de tous les articles recommandés sur lesquels l'utilisateur a cliqué. Amazon Personalize renvoie ensuite une moyenne normalisée de tous les scores des utilisateurs. Plus la valeur `average_rewards_at_k` est proche de 1, plus vous pouvez vous attendre à générer en moyenne des revenus par utilisateur grâce aux recommandations.

Pour de plus amples informations, veuillez consulter Optimisation d'une solution pour un objectif supplémentaire.

normalized_discounted_cumulative_gain_with_event_weights_at_k

Lorsque vous créez une version de solution (entraînez un modèle) pour une solution avec une configuration d'événements, Amazon Personalize génère une `normalized_discounted_cumulative_gain_with_event_weights_at_k` métrique. Le score de `normalized_discounted_cumulative_gain_with_event_weights_at_k` indique les performances de la version de la solution compte tenu de la pondération des événements que vous avez définie pour chaque type d'événement.

Il est similaire au gain cumulé actualisé normalisé (NDCG) à K, mais la récompense pour chaque prédiction correcte sera pondérée. En revanche, dans le NDCG original à K, chaque prédiction correcte aura toutes une pondération de 1. Par exemple, avec un « achat » d'un poids de 0,3 et un « clic » d'un poids de 0,1, la prédiction correcte de l'article « d'achat » donnera droit à une récompense de 1,5 tandis que la prédiction d'un article « clic » obtiendra une récompense de 0,5.

Pour de plus amples informations, veuillez consulter Optimisation d'une solution avec configuration d'événements.

précision de la prévision des tendances

Si vous avez entraîné la version de la solution avec la Tendance actuelle recette, le taux d'augmentation de la popularité des articles recommandés par le modèle. Plus la précision de la prédiction des tendances est élevée (plus elle est proche de 1), plus le modèle est capable d'identifier correctement les éléments de tendance.

Pour calculer l'accélération de la popularité, Amazon Personalize divise le taux d'augmentation de la popularité de tous les articles recommandés par l'augmentation de popularité totale des 25 articles les plus populaires. Ces éléments proviennent des interactions réelles dans le kit de test.

En fonction de la distribution de vos données et de la fréquence de découverte des tendances que vous choisissez, la valeur de précision de la prévision des tendances peut être de 0,0.

hit (hit at K)

Si vous avez entraîné la version de la solution à l'aide d'une recette USER_SEGMENTATION, le nombre moyen d'utilisateurs dans les K les plus pertinents prévus correspond aux utilisateurs réels. Les utilisateurs réels sont les utilisateurs qui ont réellement interagi avec les éléments du kit de test. K est le top 1 % des utilisateurs les plus pertinents. Plus la valeur est élevée, plus les prévisions sont précises.

rappel (rappel à K)

Si vous avez entraîné la version de la solution à l'aide d'une recette USER_SEGMENTATION, le pourcentage moyen d'utilisateurs prévus dans les K les plus pertinents prévus correspond aux utilisateurs réels. Les utilisateurs réels sont les utilisateurs qui ont réellement interagi avec les éléments du kit de test. K est le top 1 % des utilisateurs les plus pertinents. Plus la valeur est élevée, plus les prévisions sont précises.

rappel

Si vous entraînez une version de solution à l'aide de la Next-Best-Action recette, cette métrique vous indique dans quelle mesure votre version de solution détecte les actions avec lesquelles les utilisateurs interagiront.

Pour calculer`recall`, pour chaque action de votre ensemble de données, Amazon Personalize divise le nombre d'utilisateurs correctement prédits pour effectuer l'action par le nombre total d'utilisateurs qui exécutent réellement l'action dans le kit de test. Amazon Personalize calcule ensuite la moyenne de toutes les actions de votre ensemble de données.

Par exemple, si 100 utilisateurs effectuent une action dans le kit de test et qu'Amazon Personalize a prédit que 50 de ces utilisateurs effectueraient l'action, `recall` l'action est la suivante : `50 / 100 = .50` Amazon Personalize applique ensuite ce calcul à toutes les actions et renvoie la moyenne.

Zone sous la courbe (AUC)

Si vous avez entraîné la version de la solution à l'aide d'une recette PERSONALIZED_ACTIONS, la zone située sous la courbe des caractéristiques de fonctionnement du récepteur pour la version de votre solution. Cette métrique indique dans quelle mesure la version de la solution parvient à identifier correctement les actions que les utilisateurs vont entreprendre.

La courbe des caractéristiques de fonctionnement du récepteur représente les performances de la version de la solution. Il trace les taux de vrais positifs (actions correctement prédites comme pertinentes) et de faux positifs (actions incorrectement prédites comme pertinentes) à différentes valeurs de seuil. L'aire sous la courbe (AUC) est un score qui résume les performances de la version de la solution en fonction de sa courbe.

L'AUC d'une version de solution peut être comprise entre 0 et 1. Plus la valeur est proche de 1, plus le modèle est capable de prévoir les actions pertinentes pour vos utilisateurs.

Popularité des articles

Si vous avez entraîné la version de la solution à l'aide de la `aws-semantic-similarity` recette, cet indicateur mesure le niveau d'engagement qu'un article a reçu (j'aime, vues, partages, etc.) par rapport aux autres articles de votre catalogue. Plus le score de popularité de l'article est élevé (plus proche de 1), plus le nombre d'interactions entre les utilisateurs de l'article est élevé par rapport aux autres articles.

Pour calculer la popularité d'un article, Amazon Personalize compte le nombre total d'utilisateurs uniques qui ont interagi avec chaque article (par le biais d'événements tels que les vues, les clics, les achats, etc.). Le nombre d'engagement brut pour chaque élément est ensuite normalisé pour tous les éléments de votre ensemble de données, ce qui donne des scores compris entre 0 et 1. Un score proche de 0 indique un faible engagement, tandis qu'un score proche de 1 indique un engagement élevé en moyenne avec les articles du catalogue.

Article Fraîcheur

Si vous avez appris la version de la solution à l'aide de la `aws-semantic-similarity` recette, cette métrique mesure la date à laquelle un article a été ajouté à votre catalogue récemment par rapport aux autres articles. Plus le score de fraîcheur de l'article est élevé (plus il est proche de 1), plus l'article a été créé récemment.

Pour calculer la fraîcheur des articles, Amazon Personalize détermine l'âge de chaque article à l'aide du CREATION_TIMESTAMP de votre jeu de données Articles. L'âge est calculé comme la différence entre le temps d'entraînement et l'horodatage de création de l'objet. Ces âges sont ensuite normalisés pour tous les items, les items les plus récents recevant des scores plus proches de 1. Un score proche de 0 indique un plus grand nombre d'articles anciens, tandis qu'un score proche de 1 indique qu'il y a en moyenne plus d'articles récents dans le catalogue.

Similarité des articles

Si vous avez entraîné la version de la solution à l'aide de la `aws-semantic-similarity` recette, cette métrique mesure le degré de corrélation sémantique entre les éléments recommandés et vos éléments d'entrée en fonction de leur contenu, de leurs attributs et de leurs métadonnées. Plus le score de similarité des articles est élevé (plus proche de 1), plus les articles recommandés correspondent étroitement aux caractéristiques de vos articles de référence.

Pour calculer la similarité des articles, Amazon Personalize calcule les scores de similarité entre chaque article recommandé et le ou les articles d'entrée à l'aide d'intégrations sémantiques dérivées des métadonnées et des attributs des articles. Un score plus proche de 0 indique une

faible similitude sémantique entre les articles du catalogue en moyenne, tandis qu'un score plus proche de 1 indique une similitude sémantique élevée entre les articles du catalogue en moyenne. Cette métrique vous aide à comprendre dans quelle mesure le modèle capture les relations sémantiques dans vos recommandations.

Exemple

Voici un exemple simple de version de solution qui produit une liste de recommandations pour un utilisateur spécifique. Les deuxième et cinquième recommandations correspondent aux enregistrements des données de test pour cet utilisateur. Voici les recommandations pertinentes. Si `K` est défini sur 5, les métriques suivantes sont générées pour l'utilisateur.

reciprocal_rank

Calcul : 1/2

Résultat : 0,5000

normalized_discounted_cumulative_gain_at_5

Calcul : (1/log(1 + 2) + 1/log(1 + 5)) / (1/log(1 + 1) + 1/log(1 + 2))

Résultat : 0,6241

precision_at_5

Calcul : 2/5

Résultat : 0,4000

Ressources supplémentaires

Pour plus d'informations sur l'évaluation d'une version de solution à l'aide de A/B A/B tests, consultez Utiliser les tests pour mesurer l'efficacité des recommandations générées par Amazon Personalize. Pour en savoir plus sur les différents types de métriques pour les systèmes de recommandation, consultez les ressources externes suivantes :

- MRR vs MAP vs NDCG : mesures d'évaluation basées sur le classement et quand les utiliser
- Gain cumulé réduit : les indicateurs de classement que vous devez connaître

- Rappel et précision à k pour les systèmes de recommandation
- Mesures d'évaluation du classement pour les systèmes de recommandation
- Caractéristiques de fonctionnement du récepteur

Déploiement d'une version de la solution Amazon Personalize avec une campagne

Pour obtenir des recommandations en temps réel avec des ressources personnalisées, une fois que vous avez terminéCréation manuelle d'une version de solution, vous êtes prêt à déployer la version de votre solution dans le cadre d'une campagne.

Une campagne déploie une version de solution (modèle entraîné) dotée d'une capacité de transaction provisionnée pour générer des recommandations en temps réel. Après avoir créé une campagne, vous utilisez les opérations de l'GetPersonalizedRankingAPI GetRecommendations or pour obtenir des recommandations. Si vous recevez des recommandations d'articles par lots ou des segments d'utilisateurs, vous n'avez pas besoin de créer de campagne. Pour plus d'informations, consultez Obtenir des recommandations d'articles par lots ou Obtenir des segments d'utilisateurs par lots.

Lorsque vous créez une campagne, vous pouvez configurer les éléments suivants :

- Vous pouvez configurer la campagne pour qu'elle soit automatiquement mise à jour afin d'utiliser la dernière version de votre solution. Pour de plus amples informations, veuillez consulter Mises à jour automatiques des campagnes.
- Vous pouvez activer les métadonnées des éléments dans les recommandations. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.
- Vous pouvez spécifier le nombre minimum de transactions provisionnées par seconde pour la campagne. Il s'agit du débit de transaction de référence pour la campagne mise en place par Amazon Personalize. Il définit les frais de facturation minimaux pour la campagne lorsqu'elle est active. Pour de plus amples informations, veuillez consulter Nombre minimal de transactions provisionnées par seconde et auto-scaling.

Vous pouvez créer une campagne avec la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Si vous souhaitez modifier les paramètres d'une campagne existante, par exemple en activant les métadonnées dans les recommandations, vous devez

mettre à jour votre campagne. Pour de plus amples informations, veuillez consulter Mettre à jour la configuration d'une campagne Amazon Personalize.

Vous engagez des frais de campagne lorsque celle-ci est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification d'Amazon Personalize.

Rubriques

- Mises à jour automatiques des campagnes
- Nombre minimal de transactions provisionnées par seconde et auto-scaling
- Métadonnées des éléments dans les recommandations
- Création d'une campagne (console)
- Création d'une campagne (AWS CLI)
- Création d'une campagne (AWS SDKs)

Mises à jour automatiques des campagnes

Lorsque vous créez une campagne, vous pouvez activer les mises à jour automatiques de la campagne. Grâce aux mises à jour automatiques, la campagne est automatiquement mise à jour pour déployer la dernière version de votre solution entraînée automatiquement ou manuellement. Cela vous permet de tenir votre campagne à jour plus facilement.

Par exemple, si votre solution utilise une formation automatique pour créer une nouvelle version de solution tous les sept jours, votre campagne sera automatiquement mise à jour pour utiliser la dernière version de la solution pour chaque formation hebdomadaire. Si vous n'utilisez pas les mises à jour automatiques des campagnes, vous devez les mettre à jour manuellement afin de déployer le dernier modèle entraîné.

- Pour activer les mises à jour automatiques des campagnes lorsque vous créez une campagne avec la console Amazon Personalize, choisissez Mettre à jour automatiquement pour utiliser la dernière version de la solution de votre solution dans les détails de la campagne. Vous trouverez l'horodatage de la dernière mise à jour sur la page des détails de la campagne.

  Pour de plus amples informations, veuillez consulter Création d'une campagne (console).
- Pour activer les mises à jour automatiques des campagnes lorsque vous utilisez l'opération CreateCampaign API, pour le `SolutionVersionArn` paramètre, spécifiez le nom de ressource

Amazon (ARN) de votre solution au `SolutionArn/$LATEST` format. Dans le`campaignConfig`, réglez `enableMetadataWithRecommendations` sur`true`.

Pour obtenir l'horodatage de la dernière mise à jour de la campagne, vous pouvez utiliser l'opération DescribeCampaign API et vérifier les `latestCampaignUpdate` détails dans la réponse.

Pour des exemples de code expliquant comment activer les mises à jour automatiques, consultez Création d'une campagne (AWS CLI) ouCréation d'une campagne (AWS SDKs).

Nombre minimal de transactions provisionnées par seconde et auto-scaling

> **Important**
>
> Un montant élevé `minProvisionedTPS` augmentera vos coûts. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire.

Lorsque vous créez une campagne Amazon Personalize, vous pouvez spécifier le nombre minimum de transactions provisionnées par seconde (`minProvisionedTPS`) pour la campagne. Il s'agit du débit de transaction de référence pour la campagne mise en place par Amazon Personalize. Il définit les frais de facturation minimaux pour la campagne lorsqu'elle est active. Une transaction est une transaction unique `GetRecommendations` ou une `GetPersonalizedRanking` demande. La valeur par défaut `minProvisionedTPS` est 1.

Si votre TPS augmente au-delà de`minProvisionedTPS`, Amazon Personalize redimensionne automatiquement la capacité allouée à la hausse ou à la baisse, mais jamais en dessous. `minProvisionedTPS` Il y a un court délai pendant lequel la capacité augmente, ce qui peut entraîner une perte de transactions. Lorsque votre trafic diminue, la capacité revient au`minProvisionedTPS`.

Vous êtes facturé pour le TPS minimum fourni ou, si vos demandes dépassent le`minProvisionedTPS`, le TPS réel. Le TPS réel est le nombre total de demandes de recommandation que vous faites. Nous vous recommandons de commencer par un faible`minProvisionedTPS`, de suivre votre consommation à l'aide CloudWatch des statistiques Amazon, puis de l'`minProvisionedTPS`augmenter si nécessaire.

Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification d'Amazon Personalize.

Métadonnées des éléments dans les recommandations

> **Important**
>
> Si vous utilisez la recette User-Personalization-v 2 ou Personalized-Ranking-v 2, vous n'avez pas à payer de frais supplémentaires pour les métadonnées. Pour toutes les autres recettes et tous les cas d'utilisation du domaine, des frais supplémentaires vous seront facturés. Pour plus d'informations, consultez les tarifs d'Amazon Personalize.

Lorsque vous recevez des recommandations, vous pouvez demander à Amazon Personalize d'inclure les métadonnées des articles dans les résultats des recommandations. Dans votre demande, vous pouvez choisir les colonnes à inclure dans votre jeu de données Items. Amazon Personalize renvoie ces données pour chaque article dans la réponse de recommandation.

Vous pouvez utiliser les métadonnées pour enrichir les recommandations dans votre interface utilisateur, par exemple en ajoutant les genres de films aux carrousels. Vous pouvez également l'utiliser pour évaluer visuellement la qualité des recommandations. Si vous utilisez l'IA générative dans votre application, vous pouvez intégrer les métadonnées aux instructions de l'IA pour générer un contenu plus pertinent. Pour plus d'informations sur l'utilisation d'Amazon Personalize avec l'IA générative, consultezAmazon Personalize et l'IA générative.

Activation des métadonnées

Pour ajouter des métadonnées aux recommandations, vous devez disposer d'un jeu de données Items avec une colonne de métadonnées. Vous n'êtes pas obligé d'utiliser les métadonnées lors de l'entraînement. Pour plus d'informations sur la création d'un jeu de données, consultezCréation d'un schéma et d'un jeu de données. Pour plus d'informations sur la mise à jour des données, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Si vous utilisez la formule User-Personalization-v 2 ou Personalized-Ranking-v 2, les nouvelles campagnes ont automatiquement la possibilité d'inclure les métadonnées des articles dans les résultats des recommandations. Vous n'avez pas activé manuellement les métadonnées pour votre campagne. Pour toutes les autres recettes et cas d'utilisation du domaine, vous devez activer l'option de métadonnées :

- Pour activer les métadonnées avec la console Amazon Personalize, lorsque vous créez la campagne, sélectionnez Retourner les métadonnées des articles dans les résultats des recommandations dans les détails de la campagne. Pour de plus amples informations, veuillez consulter Création d'une campagne (console).
- Pour activer les métadonnées avec le AWS SDKs ou AWS CLI, utilisez l'opération CreateCampaign API et dans le `campaignConfig` set `enableMetadataWithRecommendations` to`true`. Pour plus d'informations, consultez Création d'une campagne (AWS CLI) ou Création d'une campagne (AWS SDKs).

Création d'une campagne (console)

> **Important**
>
> Vous engagez des frais de campagne lorsque celle-ci est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification d'Amazon Personalize.

Une fois que le statut de la version de votre solution est actif, vous êtes prêt à la déployer dans le cadre d'une campagne Amazon Personalize.

Pour créer une campagne (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données contenant la version de solution que vous souhaitez déployer.
3. Dans le volet de navigation, sous Ressources personnalisées, sélectionnez Campagnes.
4. Sur la page Campagnes, choisissez Créer une campagne.
5. Sur la page Créer une nouvelle campagne, pour les détails de la campagne, fournissez les informations suivantes :
   - Nom de la campagne — Entrez le nom de la campagne. Le texte que vous saisissez ici apparaît sur le tableau de bord et la page de détails de la campagne.
   - Solution — Choisissez la solution que vous venez de créer.

- Mise à jour automatique pour utiliser la dernière version de la solution de votre solution : choisissez cette option pour que la campagne utilise automatiquement la dernière version active de la solution. Si vous ne choisissez pas cette option, vous devez mettre à jour manuellement la campagne chaque fois que vous souhaitez déployer une nouvelle version de solution. Pour de plus amples informations, veuillez consulter Mises à jour automatiques des campagnes.
- ID de version de la solution : si vous n'utilisez pas les mises à jour automatiques des campagnes pour utiliser la dernière version de la solution, choisissez l'ID de la version de solution que vous souhaitez déployer.
- Minimum de transactions provisionnées par seconde (appelé MinProvisionedTPS in APIs) : définissez le nombre minimum de transactions provisionnées par seconde prises en charge par Amazon Personalize. Une valeur élevée augmentera vos frais. Nous vous recommandons de commencer par 1 (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de transactions provisionnées par seconde et auto-scaling.
- Renvoyer les métadonnées des éléments dans les résultats des recommandations : choisissez cette option si vous souhaitez inclure les métadonnées dans les résultats des recommandations. Si cette option est activée, vous pouvez spécifier les colonnes de votre jeu de données Items lorsque vous recevez des recommandations. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.

6. Si vous avez utilisé la recette de personnalisation par l'utilisateur, dans la configuration de la campagne, vous pouvez éventuellement saisir des valeurs pour le poids d'exploration et la limite d'âge des objets d'exploration. Pour de plus amples informations, veuillez consulter Personnalisation par l'utilisateur.
7. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
8. Choisissez Create campaign (Créer une campagne).
9. Sur la page des détails de la campagne, lorsque le statut de la campagne est Actif, vous pouvez utiliser la campagne pour obtenir des recommandations et enregistrer des impressions. Pour de plus amples informations, veuillez consulter Obtenir des recommandations d'Amazon Personalize.

   La campagne est prête lorsque son statut est ACTIF. Si vous modifiez la version de votre solution ou si vous souhaitez modifier les paramètres de votre campagne, vous devez mettre

à jour votre campagne. Pour de plus amples informations, veuillez consulter Mettre à jour la configuration d'une campagne Amazon Personalize.

Création d'une campagne (AWS CLI)

> **Important**
>
> Vous engagez des frais de campagne lorsque celle-ci est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification d'Amazon Personalize.

Une fois que la version de votre solution est active, vous êtes prêt à la déployer dans le cadre d'une campagne Amazon Personalize. Pour créer une campagne avec le AWS CLI, vous devez utiliser la `create-campaign` commande.

L'exemple de code suivant vous montre comment créer une campagne. Il déploie la dernière version d'une solution qui utilise la recette de personnalisation utilisateur. La campagne qu'il crée est automatiquement mise à jour pour utiliser les futures versions de la solution. Le code utilise la configuration suivante :

- Il configure la campagne pour qu'elle soit automatiquement mise à jour afin d'utiliser la dernière version de la solution pour votre solution : elle `solution-version-arn` est au *`solution ARN`*`/$LATEST` format, et elle l'`syncWithLatestSolutionVersion`est`True`. Pour utiliser le code, remplacez-le `solution ARN` par le Amazon Resource Name (ARN) de votre solution.

  Pour désactiver la fonction automatique`syncWithLatestSolutionVersion`, spécifiez uniquement l'ARN de la version de la solution (sans`/$LATEST`) et définissez `syncWithLatestSolutionVersion` sur`False`.
- Il définit l'`enableMetadataWithRecommendations`option sur`True`. Cela permet à une option de demande de recommandation d'inclure les métadonnées d'un élément d'un jeu de données d'articles avec les résultats des recommandations. Pour désactiver cette option, définissez-la sur`False`. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.
- Il est défini `min-provisioned-tps` sur 1 (valeur par défaut). Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre consommation

à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de transactions provisionnées par seconde et auto-scaling.

Pour obtenir la liste complète de tous les paramètres, voirCreateCampaign.

```
aws personalize create-campaign \
--name campaign-name \
--solution-version-arn solution-arn/\$LATEST \
--min-provisioned-tps 1 \
--campaign-config "{"\"syncWithLatestSolutionVersion"\": "true",
 "\"enableMetadataWithRecommendations"\": "true"}"
```

La campagne est prête lorsque son statut est ACTIF. Pour obtenir l'état actuel, appelez l'API DescribeCampaign et vérifiez que le champ `status` est `ACTIVE`.

Si vous modifiez la version de votre solution et que votre campagne n'est pas automatiquement mise à jour pour utiliser la dernière version de la solution, ou si vous souhaitez modifier les paramètres de votre campagne, vous devez mettre à jour votre campagne. Pour de plus amples informations, veuillez consulter Mettre à jour la configuration d'une campagne Amazon Personalize.

Amazon Personalize vous propose des opérations de gestion de campagnes, telles que ListCampaigns la liste des campagnes que vous avez créées. Vous pouvez supprimer une campagne en appelant l'API DeleteCampaign. Si vous supprimez une campagne, les versions des solutions qui font partie de cette campagne ne sont pas supprimées.

Après avoir créé votre campagne, vous pouvez l'utiliser pour faire des recommandations. Pour de plus amples informations, veuillez consulter Obtenir des recommandations d'Amazon Personalize.

Création d'une campagne (AWS SDKs)

> **Important**
>
> Vous engagez des frais de campagne lorsque celle-ci est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification d'Amazon Personalize.

Une fois que la version de votre solution est active, vous êtes prêt à la déployer dans le cadre d'une campagne Amazon Personalize. Pour créer une campagne avec AWS SDKs, vous devez utiliser l'opération CreateCampaign API.

L'exemple de code suivant vous montre comment créer une campagne. Le code déploie la dernière version d'une solution qui utilise la recette de personnalisation utilisateur. La campagne qu'il crée est automatiquement mise à jour pour utiliser les futures versions de la solution. Le code utilise la configuration suivante :

- Il configure la campagne pour qu'elle soit automatiquement mise à jour afin d'utiliser la dernière version de la solution pour votre solution : elle `solutionVersionArn` est au *`solution ARN`*`/$LATEST` format, et elle l'`syncWithLatestSolutionVersion`est`True`. Pour utiliser le code, remplacez-le `solution ARN` par le Amazon Resource Name (ARN) de la version de votre solution.

  Pour désactiver la fonction automatique`syncWithLatestSolutionVersion`, spécifiez uniquement l'ARN de la version de la solution (sans`/$LATEST`) et définissez `syncWithLatestSolutionVersion` sur`False`.
- Il définit l'`enableMetadataWithRecommendations`option sur`True`. Cela permet à une option de demande de recommandation d'inclure les métadonnées d'un élément d'un jeu de données d'articles avec les résultats des recommandations. Pour désactiver cette option, définissez-la sur`False`. Pour de plus amples informations, veuillez consulter Métadonnées des éléments dans les recommandations.
- Il est défini `minProvisionedTPS` sur 1 (valeur par défaut). Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire. Pour de plus amples informations, veuillez consulter Nombre minimal de transactions provisionnées par seconde et auto-scaling.

Pour obtenir la liste complète de tous les paramètres, voirCreateCampaign.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_campaign(
```

```
    name = 'campaign name',
    solutionVersionArn = 'solution ARN/$LATEST',
    minProvisionedTPS = 1,
    campaignConfig = {"syncWithLatestSolutionVersion": True,
 "enableMetadataWithRecommendations": True}
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateCampaignCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the campaign parameters
export const createCampaignParam = {
  solutionVersionArn: "SOLUTION_ARN/$LATEST" /* required */,
  name: "NAME" /* required */,
  minProvisionedTPS: 1 /* optional */,
  campaignConfig: {   /* optional */
    syncWithLatestSolutionVersion: true,
    enableMetadataWithRecommendations: true,
  },
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateCampaignCommand(createCampaignParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
```

```
  }
};
run();
```

La campagne est prête lorsque son statut est ACTIF. Pour connaître le statut actuelDescribeCampaign, appelez et vérifiez que le `status` champ est bien renseigné`ACTIVE`.

Si vous modifiez manuellement la version de votre solution ou si vous souhaitez modifier les paramètres de votre campagne, vous devez mettre à jour votre campagne. Pour de plus amples informations, veuillez consulter Mettre à jour la configuration d'une campagne Amazon Personalize.

Amazon Personalize vous propose des opérations de gestion de campagnes, telles que ListCampaigns la liste des campagnes que vous avez créées. Vous pouvez supprimer une campagne en appelant l'API DeleteCampaign. Si vous supprimez une campagne, les versions des solutions qui font partie de cette campagne ne sont pas supprimées.

Une fois que vous avez créé votre campagne, utilisez-la pour effectuer des recommandations. Pour de plus amples informations, veuillez consulter Obtenir des recommandations d'Amazon Personalize.

Mettre à jour la configuration d'une campagne Amazon Personalize

Pour modifier votre campagneTPS minimum provisionné, déployer manuellement une nouvelle version de solution ou modifier la configuration d'une campagne, par exemple en activant l'option permettant d'inclure des métadonnées dans les recommandations, vous devez mettre à jour manuellement la campagne.

Les actions suivantes ne nécessitent pas de mise à jour manuelle de la campagne :

- Si votre campagne utilise des mises à jour automatiques, vous n'êtes pas obligé de la mettre à jour pour déployer la dernière version de votre solution entraînée automatiquement ou manuellement. Pour plus d’informations, consultez Mises à jour automatiques des campagnes.
- Avec User-Personalization-v2, ou User-Personalization Next-Best-Action, Amazon Personalize met automatiquement à jour la dernière version de votre solution toutes les deux heures pour inclure de nouveaux éléments ou actions dans les recommandations. Votre campagne utilise automatiquement la version mise à jour de la solution.

Vous mettez à jour manuellement une campagne à l'aide de la console Amazon Personalize AWS Command Line Interface (AWS CLI) ou AWS des SDK.

> **Note**
>
> Pour mettre à jour une campagne afin d'activer l'option permettant d'inclure des métadonnées dans les recommandations, vous devez disposer d'un jeu de données Items avec une colonne de métadonnées. Vous n'êtes pas obligé d'utiliser les métadonnées lors de l'entraînement.
> Si votre campagne a précédemment déployé une version de solution qui utilisait User-Personalization-v2 ou Personalized-Ranking-v2, et que vous passez à une ancienne version des recettes, l'option d'inclusion des métadonnées est désactivée par défaut. Vous pouvez l'activer lorsque vous mettez à jour la campagne. Pour plus d'informations, consultez Métadonnées des éléments dans les recommandations.

Rubriques

- Mettre à jour une campagne (console)
- Mettre à jour une campagne (AWS CLI)
- Mettre à jour une campagne (AWS SDK)

Mettre à jour une campagne (console)

Pour déployer une version de solution reformatée manuellement ou apporter des modifications à la configuration de votre campagne, vous devez mettre à jour votre campagne.

Pour mettre à jour une campagne (console)

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données contenant la campagne que vous souhaitez mettre à jour.
3. Dans le volet de navigation, choisissez Campaigns (Campagnes).
4. Sur la page Campagnes, choisissez la campagne que vous souhaitez mettre à jour.
5. Sur la page des détails de la campagne, choisissez Mettre à jour.
6. Sur la page Mettre à jour la campagne, apportez vos modifications. Par exemple, si vous déployez une version de solution recyclée, choisissez le numéro d'identification de la nouvelle version de solution dans le champ ID de version de solution.

7. Choisissez Mettre à jour. Amazon Personalize met à jour la campagne afin d'utiliser la nouvelle version de la solution et toutes les configurations modifiées.

Mettre à jour une campagne (AWS CLI)

Pour déployer une nouvelle version de solution, modifier celle de TPS minimum provisionné votre campagne ou modifier la configuration de votre campagne, vous devez mettre à jour votre campagne. Utilisez la `update-campaign` commande suivante pour mettre à jour une campagne afin d'utiliser une nouvelle version de solution avec le AWS CLI.

`campaign arn`Remplacez-le par le Amazon Resource Name (ARN) de la campagne que vous souhaitez mettre à jour. `new solution version arn`Remplacez-le par la version de solution que vous souhaitez déployer.

```
aws personalize update-campaign \
--campaign-arn campaign arn \
--solution-version-arn new solution version arn \
--min-provisioned-tps 1
```

Mettre à jour une campagne (AWS SDK)

Pour déployer une nouvelle version de solution, modifier votre campagne TPS minimum provisionné ou modifier la configuration de votre campagne, vous devez mettre à jour votre campagne. Utilisez le code suivant pour mettre à jour une campagne avec le AWS SDK pour Python (Boto3) ou le SDK for Java 2.x. Pour obtenir une liste complète de paramètres, consultez UpdateCampaign.

SDK for Python (Boto3)

Utilisez la `update_campaign` méthode suivante pour déployer une nouvelle version de solution. `campaign arn`Remplacez-le par le nom de ressource Amazon (ARN) de la campagne que vous souhaitez mettre à jour, remplacez le `new solution version arn` par l'ARN de la nouvelle version de la solution et modifiez éventuellement le`minProvisionedTPS`.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.update_campaign(
    campaignArn = 'campaign arn',
```

```
    solutionVersionArn = 'new solution version arn',
    minProvisionedTPS = 1,
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

Utilisez la `updateCampaign` méthode suivante pour mettre à jour une campagne afin d'utiliser une nouvelle version de solution. Transmettez comme paramètres un client du service Amazon Personalize, le nom de ressource Amazon (ARN) de la nouvelle version de la solution et leTPS minimum provisionné.

```
public static void updateCampaign(PersonalizeClient personalizeClient,
                                  String campaignArn,
                                  String solutionVersionArn,
                                  Integer minProvisionedTPS) {

    try {
        // build the updateCampaignRequest
        UpdateCampaignRequest updateCampaignRequest =
 UpdateCampaignRequest.builder()
            .campaignArn(campaignArn)
            .solutionVersionArn(solutionVersionArn)
            .minProvisionedTPS(minProvisionedTPS)
            .build();

        // update the campaign
        personalizeClient.updateCampaign(updateCampaignRequest);

        DescribeCampaignRequest campaignRequest = DescribeCampaignRequest.builder()
              .campaignArn(campaignArn)
              .build();

        DescribeCampaignResponse campaignResponse =
 personalizeClient.describeCampaign(campaignRequest);
        Campaign updatedCampaign = campaignResponse.campaign();
```

```
        System.out.println("The Campaign status is " + updatedCampaign.status());

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Obtenir des recommandations d'Amazon Personalize

Après avoir créé un recommandeur ou créé une campagne, vous êtes prêt à recevoir des recommandations. En fonction de vos ressources, vous pouvez obtenir des recommandations en temps réel ou via un flux de travail par lots.

- Grâce aux ressources personnalisées, vous pouvez obtenir des recommandations en temps réel ou des recommandations par lots. Pour obtenir des recommandations en temps réel, vous devez créer une campagne personnalisée avant de recevoir des recommandations. Pour les recommandations par lots, il n'est pas nécessaire de créer de campagne.
- Lorsque des recommandations figurent dans un groupe de données de domaine, vous ne pouvez obtenir que des recommandations en temps réel.

Les rubriques suivantes expliquent comment et quand utiliser chaque type de recommandation. Grâce aux recommandations par lots et en temps réel, vous pouvez filtrer les résultats. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Rubriques

- Scores de recommandation
- Recommandations d'articles en temps réel dans Amazon Personalize
- Recommandations d'action en temps réel dans Amazon Personalize
- Obtenir un classement personnalisé (ressources personnalisées)
- Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles
- Obtenir des recommandations d'articles par lots grâce à des ressources personnalisées
- Obtenir des segments d'utilisateurs par lots avec des ressources personnalisées

Scores de recommandation

Grâce aux solutions personnalisées créées à partir des recettes User-Personalization-v 2, User-Personalization, Personalized-Ranking-v 2, Personalized-Ranking et PERSONALIZED_ACTIONS, Amazon Personalize inclut un score pour chaque article dans les recommandations. Ces scores représentent la certitude relative dont dispose Amazon Personalize quant à l'article ou à l'action que l'utilisateur sélectionnera ensuite. Des scores plus élevés représentent une plus grande certitude.

- Pour plus d'informations sur les scores de User-Personalization-v 2 et sur la personnalisation par l'utilisateur, consultez. Comment fonctionne la notation des recommandations (ressources personnalisées)
- Pour plus d'informations sur les scores des recettes PERSONALIZED_ACTIONS, consultez. Comment fonctionne la notation des recommandations d'action
- Pour plus d'informations sur les scores pour Personalized-Ranking-v 2 et les recommandations relatives au classement personnalisé, voir. Comment fonctionne le classement personnalisé

Pour les tâches d'inférence par lots, les scores des articles sont calculés comme décrit dans Comment fonctionne la notation des recommandations (ressources personnalisées) etComment fonctionne le classement personnalisé. Vous pouvez consulter les scores dans le fichier JSON de sortie de la tâche d'inférence par lots.

Recommandations d'articles en temps réel dans Amazon Personalize

Si votre cas d'utilisation ou votre recette génère des recommandations d'articles, après avoir créé un outil de recommandation ou créé une campagne, vous pouvez obtenir des recommandations d'articles personnalisées ou connexes en temps réel pour vos utilisateurs.

Si le cas d'utilisation ou la recette de votre domaine propose une personnalisation en temps réel, comme le cas d'utilisation Top picks for you ou la recette User-Personalization-v2, Amazon Personalize met à jour les recommandations en fonction de l'activité la plus récente de votre utilisateur lorsque vous enregistrez ses interactions avec votre catalogue. Pour plus d'informations sur l'enregistrement des événements en temps réel et la personnalisation, consultezEnregistrement des événements en temps réel pour influencer les recommandations.

Lorsque vous recevez des recommandations d'articles en temps réel, vous pouvez effectuer les opérations suivantes :

- Si vous avez configuré votre campagne pour renvoyer des métadonnées pour les éléments recommandés, vous pouvez spécifier les colonnes à inclure dans votre opération GetRecommendations d'API. Vous pouvez également spécifier les colonnes lorsque vous testez la campagne avec la console Amazon Personalize. Pour obtenir des exemples de code, veuillez consulter Obtenir les métadonnées des articles avec des recommandations en temps réel. Pour plus d'informations sur l'activation des métadonnées pour une campagne,

consultezMétadonnées des éléments dans les recommandations. Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultezActivation des métadonnées dans les recommandations pour un recommandeur de domaine dans Amazon Personalize.

- Pour certains cas d'utilisation et certaines recettes, vous pouvez spécifier une promotion dans votre demande de recommandation. Une promotion définit des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés. Pour de plus amples informations, veuillez consulter Promouvoir des articles dans le cadre de recommandations en temps réel.
- Vous pouvez filtrer les résultats en fonction de critères personnalisés. Par exemple, il se peut que vous ne souhaitiez pas recommander des produits qu'un utilisateur a déjà achetés ou recommander uniquement des articles destinés à un groupe d'âge en particulier. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

> **Note**
>
> Si vous avez utilisé une recette personnalisée PERSONALIZED_RANKING, consultez. Obtenir un classement personnalisé (ressources personnalisées)

Rubriques

- Comment fonctionne la notation des recommandations (ressources personnalisées)
- Motifs de recommandation avec User-Personalization-v 2
- Obtenir des recommandations d'articles en temps réel
- Obtenir les métadonnées des articles avec des recommandations en temps réel
- Promouvoir des articles dans le cadre de recommandations en temps réel

Comment fonctionne la notation des recommandations (ressources personnalisées)

Avec les recettes User-Personalization-v 2 et User-Personalization, Amazon Personalize génère des scores pour les articles en fonction des données d'interaction et des métadonnées de l'utilisateur. Ces scores représentent la certitude relative dont dispose Amazon Personalize quant à savoir si l'utilisateur interagira ensuite avec l'article. Des scores plus élevés représentent une plus grande certitude.

> **Note**
>
> Amazon Personalize n'affiche pas les scores des recommandations de domaines ou des recettes Similar-Items, SIMS ou Popularity-Count. Pour plus d'informations sur les scores des recommandations de classement personnalisé, voir. Comment fonctionne le classement personnalisé

Amazon Personalize génère des scores pour les articles les uns par rapport aux autres sur une échelle de 0 à 1 (les deux inclus). Avec User-Personalization-v 2, Amazon Personalize génère des scores pour un sous-ensemble de vos articles. Grâce à la personnalisation par l'utilisateur, Amazon Personalize note tous les articles de votre catalogue.

Si vous utilisez User-Personalization-v 2 et que vous appliquez un filtre aux recommandations, en fonction du nombre de recommandations supprimées par le filtre, Amazon Personalize peut ajouter des éléments réservés. Il le fait pour répondre à `numResults` votre demande de recommandation. Ces éléments sont des éléments populaires, basés sur le nombre de données d'interactions, qui répondent à vos critères de filtre. Ils n'ont pas de score de pertinence pour l'utilisateur.

Pour User-Personalization-v 2 et pour la personnalisation par l'utilisateur, le total de tous les scores est égal à 1. Par exemple, si vous recevez des recommandations de films pour un utilisateur et que trois films apparaissent dans le jeu de données Items et le jeu de données Interactions, leurs scores peuvent être `0.60.3`, et`0.1`. De même, si vous avez 10 000 films dans votre inventaire, les films les mieux notés peuvent avoir des scores très faibles (le score moyen serait le cas`.001`), mais comme le score est relatif, les recommandations restent valables.

En termes mathématiques, les scores de chaque paire utilisateur-élément (u, i) sont calculés selon la formule suivante, où `exp` est la fonction exponentielle, w_u et w_i/$_j$sont respectivement des intégrations utilisateur et élément, et la lettre grecque sigma (Σ) représente la somme de tous les éléments avec des scores :

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_j \exp(\bar{w}_u^\top w_j)}$$

Motifs de recommandation avec User-Personalization-v 2

Si vous en utilisez User-Personalization-v 2, les éléments que le modèle ne recommande normalement pas incluent une `reason` liste. Ces raisons expliquent pourquoi le point a été inclus dans les recommandations. Les raisons possibles sont les suivantes :

- Article sponsorisé : indique que l'article a été inclus dans le cadre d'une promotion que vous avez appliquée dans votre demande de recommandation.
- Exploration — Indique que l'objet a été inclus dans l'exploration. Dans le cadre de l'exploration, les recommandations incluent des éléments présentant moins de données d'interactions ou de pertinence pour l'utilisateur. Pour plus d'informations sur l'exploration, voir Exploration.
- Article populaire — Indique que l'article a été inclus en tant qu'objet populaire de remplacement. Si vous utilisez un filtre, en fonction du nombre de recommandations qu'il supprime, Amazon Personalize peut ajouter des éléments réservés `numResults` pour répondre à votre demande de recommandation. Ces éléments sont des éléments populaires, basés sur les données d'interactions, qui répondent à vos critères de filtre. Ils n'ont pas de score de pertinence pour l'utilisateur.

Obtenir des recommandations d'articles en temps réel

Vous pouvez obtenir des recommandations d'articles en temps réel à partir d'un outil de recommandation Amazon Personalize ou d'une campagne personnalisée à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI), ou AWS SDKs.

Rubriques

- Obtenir des recommandations d'articles (console)
- Obtenir des recommandations d'articles (AWS CLI)
- Obtenir des recommandations d'articles (AWS SDKs)

Obtenir des recommandations d'articles (console)

Pour obtenir des recommandations à l'aide de la console Amazon Personalize, vous devez fournir les informations relatives à la demande sur la page détaillée d'un recommandeur (groupe de données de domaines) ou d'une campagne personnalisée.

Obtention de recommandations

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne ou le recommandateur que vous utilisez.
3. Dans le volet de navigation, choisissez Campaigns ou Recommenders.
4. Choisissez la campagne cible ou le recommandateur.
5. Pour une campagne, sous Tester les résultats de la campagne, saisissez les détails de votre demande de recommandation en fonction de la recette que vous avez utilisée. Pour un recommandateur, choisissez Test recommender et entrez les détails de votre demande de recommandation en fonction de votre cas d'utilisation.

   Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.
6. Choisissez éventuellement un filtre. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.
7. Si vous utilisez des métadonnées contextuelles, fournissez des données pour chaque contexte. Pour chaque contexte, saisissez le champ de métadonnées pour la clé. Pour la valeur, entrez les données contextuelles. Pour de plus amples informations, veuillez consulter Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles.
8. Si vous avez activé les métadonnées dans les recommandations pour votre campagne ou votre recommandateur, pour les colonnes du jeu de données Items, choisissez les colonnes de métadonnées que vous souhaitez inclure dans les résultats des recommandations. Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultezMétadonnées des éléments dans les recommandations. Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultezActivation des métadonnées dans les recommandations pour un recommandeur de domaine dans Amazon Personalize.
9. Si vous souhaitez promouvoir un sous-ensemble d'articles, complétez éventuellement les champs de promotion. Pour de plus amples informations, veuillez consulter Promouvoir des articles dans le cadre de recommandations en temps réel.

10. Choisissez Get recommendations (Obtenir des recommandations). Un tableau contenant les 25 principaux éléments recommandés par l'utilisateur s'affiche. Si vous utilisez User-Personalization-v 2, chaque élément recommandé inclut une liste des raisons pour lesquelles l'élément a été inclus dans les recommandations. Pour de plus amples informations, veuillez consulter Motifs de recommandation avec User-Personalization-v 2.

Obtenir des recommandations d'articles (AWS CLI)

Utilisez le code suivant pour obtenir les recommandations d'une campagne. Pour obtenir les recommandations d'un recommandeur, remplacez le `campaign-arn` paramètre par. `recommender-arn`

Spécifiez l'ID de l'utilisateur pour lequel vous souhaitez obtenir des recommandations, ainsi que le nom de ressource Amazon (ARN) de votre campagne ou de votre recommandateur. La liste des 10 éléments les plus recommandés pour l'utilisateur s'affiche. Si vous utilisez User-Personalization-v 2, chaque élément recommandé inclut une liste des raisons pour lesquelles l'élément a été inclus dans les recommandations. Pour de plus amples informations, veuillez consulter Motifs de recommandation avec User-Personalization-v 2.

Pour modifier le nombre d'éléments recommandés, modifiez la valeur de`numResults`. La valeur par défaut est de 25 éléments. Le maximum est de 500 articles. Si vous avez utilisé une recette RELATED_ITEMS pour entraîner la version de solution soutenant la campagne, remplacez le `user-id` paramètre par `item-id` et spécifiez l'ID de l'article.

Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

Obtenir des recommandations d'articles (AWS SDKs)

Le code suivant montre comment obtenir des recommandations Amazon Personalize pour un utilisateur à partir d'une campagne avec le AWS SDKs. Pour obtenir les recommandations d'un recommandeur, remplacez le `campaignArn` paramètre par. `recommenderArn`

Spécifiez l'ID de l'utilisateur pour lequel vous souhaitez obtenir des recommandations, ainsi que le nom de ressource Amazon (ARN) de votre campagne ou de votre recommandateur. La liste des 10 éléments les plus recommandés pour l'utilisateur s'affiche. Si vous utilisez User-Personalization-v 2, chaque élément recommandé inclut une liste des raisons pour lesquelles l'élément a été inclus dans les recommandations. Pour de plus amples informations, veuillez consulter Motifs de recommandation avec User-Personalization-v 2.

Pour modifier le nombre d'éléments recommandés, modifiez la valeur de`numResults`. La valeur par défaut est de 25 éléments. Le maximum est de 500 articles. Si vous avez utilisé une recette RELATED_ITEMS pour entraîner la version de solution soutenant la campagne, remplacez le `userId` paramètre par `itemId` et spécifiez l'ID de l'article.

Si vous avez activé les métadonnées dans les recommandations pour votre campagne ou votre recommandateur, vous pouvez spécifier les colonnes de métadonnées du jeu de données Items à inclure dans la réponse. Pour obtenir un exemple de code, consultez Inclure les métadonnées des éléments avec les recommandations (AWS SDKs). Pour plus d'informations sur l'activation des métadonnées, consultezMétadonnées des éléments dans les recommandations.

Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)
```

```
print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

SDK for Java 2.x

```
    public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
 String campaignArn, String userId) {

        try {
            GetRecommendationsRequest recommendationsRequest =
 GetRecommendationsRequest.builder()
                    .campaignArn(campaignArn)
                    .numResults(20)
                    .userId(userId)
                    .build();

            GetRecommendationsResponse recommendationsResponse =
 personalizeRuntimeClient
                    .getRecommendations(recommendationsRequest);
            List<PredictedItem> items = recommendationsResponse.itemList();
            for (PredictedItem item : items) {
                System.out.println("Item Id is : " + item.itemId());
                System.out.println("Item score is : " + item.score());
            }

        } catch (AwsServiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";

import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});
```

```
// Set the recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Obtenir les métadonnées des articles avec des recommandations en temps réel

Si vous avez configuré votre campagne ou votre recommandation pour renvoyer des métadonnées pour les éléments recommandés, vous pouvez spécifier les colonnes à inclure dans votre opération GetRecommendations d'API. Vous pouvez également spécifier les colonnes lorsque vous testez la campagne avec la console Amazon Personalize.

Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultezMétadonnées des éléments dans les recommandations. Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultezActivation des métadonnées dans les recommandations pour un recommandeur de domaine dans Amazon Personalize.

Les exemples de code suivants montrent comment spécifier les colonnes de métadonnées à inclure dans le AWS CLI ou le AWS SDKs. Pour ce faire, utilisez la console Amazon Personalize, vous devez spécifier les colonnes lorsque vous testez votre campagne. Pour de plus amples informations, veuillez consulter Obtenir des recommandations d'articles (console).

Rubriques

- Inclure les métadonnées des éléments avec les recommandations (AWS CLI)
- Inclure les métadonnées des éléments avec les recommandations (AWS SDKs)

Inclure les métadonnées des éléments avec les recommandations (AWS CLI)

Si vous avez activé les métadonnées dans les recommandations pour votre campagne ou votre recommandateur, vous pouvez spécifier les colonnes de métadonnées du jeu de données Items à inclure dans la réponse. L'exemple de code suivant montre comment spécifier les colonnes de métadonnées dans le cadre de votre demande de recommandations.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10 \
--metadata-columns "{\"ITEMS\": [\"\"columnNameA\"\",\"\"columnNameB\"\"]}"
```

Inclure les métadonnées des éléments avec les recommandations (AWS SDKs)

Si vous avez activé les métadonnées dans les recommandations pour votre campagne ou votre recommandateur, vous pouvez spécifier les colonnes de métadonnées du jeu de données Items à inclure dans la réponse. L'exemple de code suivant montre comment spécifier les colonnes de métadonnées dans le cadre de votre demande de recommandations.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
  campaignArn = 'Campaign ARN',
  userId = 'User ID',
  numResults = 10
  metadataColumns = {
    "ITEMS": ['columnNameA','columnNameB']
  }
)

print("Recommended items")
for item in response['itemList']:
  print(item['itemId'])
  print(item['metadata'])
```

Promouvoir des articles dans le cadre de recommandations en temps réel

Avec tous les cas d'utilisation de domaines et certaines recettes personnalisées, vous pouvez définir une promotion lorsque vous recevez des recommandations en temps réel.

Une promotion définit des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés. Par exemple, vous avez peut-être une application de streaming et souhaitez promouvoir vos propres émissions et films, mais également recommander des titres pertinents. Vous pouvez utiliser une promotion pour spécifier qu'un certain pourcentage d'articles recommandés doivent provenir de la catégorie en interne. Les autres articles recommandés continueraient d'être des recommandations pertinentes en fonction de votre recette et de tout filtre de demande.

Pour appliquer une promotion, vous devez spécifier les éléments suivants dans votre demande de recommandation :

- Pourcentage d'articles recommandés auxquels appliquer le filtre de promotion.
- Filtre qui définit les critères de promotion. Pour plus d’informations, consultez Filtres de promotion.

Dans la réponse à la recommandation, les articles promus sont positionnés de manière aléatoire par rapport aux autres articles recommandés, mais dans un ordre trié par rapport aux autres articles promus. En fonction de votre recette, les articles recommandés qui ne font pas partie d'une promotion sont triés en fonction de leur pertinence pour l'utilisateur, de leur popularité ou de leur similitude. S'il n'y a pas assez d'articles répondant aux critères de promotion, le résultat contiendra autant d'articles sponsorisés que possible.

Vous pouvez appliquer une promotion aux recommandations à l'aide de la console Amazon Personalize AWS Command Line Interface (AWS CLI) ou AWS des SDK.

Rubriques

- Cas d'utilisation et recettes qui soutiennent les promotions
- Filtres de promotion
- Promouvoir de nouveaux articles
- Promouvoir des articles (console)
- Articles de promotion (AWS CLI)
- Articles de promotion (AWS SDK)

Cas d'utilisation et recettes qui soutiennent les promotions

Tous les cas d'utilisation prennent en charge les promotions. Les recettes personnalisées suivantes soutiennent les promotions :

- Personnalisation de l'utilisateur-v2et Personnalisation par l'utilisateur recettes
- Objets similairesset SIMS recettes
- Tendance actuelleet Comptage de popularité recettes

Filtres de promotion

Lorsque vous appliquez une promotion à une demande de recommandation, vous choisissez un filtre qui précise les critères de promotion. Vous pouvez utiliser un filtre existant ou en créer un nouveau. Vous créez et gérez des filtres pour les promotions comme vous le feriez avec d'autres filtres dans Amazon Personalize. Pour plus d'informations sur la création et la gestion des filtres, consultezFiltrer les résultats.

La seule différence entre un filtre de promotion et un filtre que vous choisissez en dehors de la promotion (le filtre de demande) réside dans la manière dont Amazon Personalize les applique. Un filtre de promotion s'applique uniquement aux articles promus, tandis qu'un filtre de demande s'applique uniquement aux articles recommandés restants. Si vous spécifiez un filtre de demande et un filtre de promotion et que vous souhaitez appliquer les deux filtres aux articles promus, l'expression de votre filtre de promotion doit inclure les deux expressions. La façon dont vous combinez deux expressions dépend des ensembles de données que vous utilisez. Pour plus d'informations sur les expressions de filtre, leurs règles et la façon de les créer, consultezExpressions de filtrage.

Exemples d'expressions de filtre

L'expression suivante inclut uniquement les éléments de la catégorie « interne ». Vous pouvez utiliser cette expression si vous souhaitez promouvoir votre propre contenu dans vos recommandations.

```
INCLUDE ItemID WHERE Items.OWNER IN ("in-house")
```

L'expression suivante inclut uniquement les éléments créés plus récemment que l'horodatage que vous spécifiez. Vous pouvez utiliser cette expression pour promouvoir de nouveaux éléments dans les recommandations.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

L'expression suivante montre comment appliquer un filtre de demande aux articles promus. Il inclut uniquement les vêtements disponibles en tant qu'articles promotionnels. Dans ce scénario, le `Items.AVAILABLE IN ("True")` serait également utilisé dans l'expression du filtre de demande, de sorte que toutes les recommandations concernent les éléments disponibles.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") AND Items.AVAILABLE IN ("True")
```

Pour une liste plus complète des exemples de filtres, voirExemples d'expressions de filtre.

Promouvoir de nouveaux articles

Si vous utilisez leRecette de personnalisation par l'utilisateur V2, Amazon Personalize recommande les articles les plus pertinents aux utilisateurs et recommande plus fréquemment les articles existants contenant des données d'interactions. Pour vous assurer que les recommandations incluent de nouveaux articles, vous pouvez appliquer une promotion aux demandes de recommandation qui inclut des articles en fonction de l'horodatage de création.

Si vous n'utilisez pas encore de promotion, votre expression de filtre peut promouvoir les articles créés après une certaine date :

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

Si vous utilisez déjà une promotion, vous créez une expression qui contient à la fois les déclarations relatives à la promotion et à l'état du nouvel article :

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") OR Items.CREATION_TIMESTAMP > $DATE
```

Promouvoir des articles (console)

Pour promouvoir certains articles dans les recommandations à l'aide de la console Amazon Personalize, créez un filtre, puis fournissez les détails de la promotion dans la demande de recommandation. Pour plus d'informations sur les autres champs, voirObtenir des recommandations d'articles (console).

Pour promouvoir des articles dans les recommandations

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne ou le recommandateur que vous utilisez.
3. Si ce n'est pas déjà fait, créez un filtre qui précise les critères de promotion. Vous créez des filtres pour les promotions de la même manière que vous créez des filtres de demande. Pour plus d'informations sur la création et la gestion des filtres, consultezFiltrer les résultats.
4. Dans le volet de navigation, choisissez Recommenders ou Campaigns.
5. Choisissez la campagne cible ou le recommandateur.
6. Pour les campagnes, sous Tester les résultats des campagnes, saisissez les détails de votre demande de recommandation en fonction de la recette que vous avez utilisée. Pour les recommandateurs, choisissez Test recommender et entrez les détails de votre demande de recommandation.
7. Choisissez éventuellement un filtre pour la demande. Ce filtre s'applique uniquement aux articles non sponsorisés. Pour plus d'informations sur la création et la gestion des filtres, consultezFiltrer les résultats.
8. Si vous utilisez des métadonnées contextuelles, fournissez des données pour chaque contexte. Pour chaque contexte, saisissez le champ de métadonnées pour la clé. Pour la valeur, entrez les données contextuelles. Pour plus d'informations, consultez Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles.
9. Pour la promotion, spécifiez les éléments suivants :

   - Pourcentage d'articles promus : entrez le pourcentage d'articles recommandés auxquels appliquer la promotion.
   - Filtre : Choisissez un filtre qui spécifie les critères de promotion. Ce filtre s'applique aux éléments promus au lieu de tout filtre de demande que vous auriez spécifié à l'étape 7.
   - Paramètre de filtre : si votre promotion utilise un filtre avec des paramètres d'espace réservé, entrez la valeur pour chaque paramètre afin de définir les critères du filtre. Pour utiliser plusieurs valeurs pour un paramètre, séparez chaque valeur par une virgule.

10. Choisissez Obtenir des recommandations. Un tableau contenant les 25 principaux éléments recommandés par l'utilisateur s'affiche. La colonne Article sponsorisé indique si l'article a été inclus en raison de votre promotion. Les articles sponsorisés sont positionnés de manière aléatoire par rapport aux autres articles recommandés, mais dans un ordre trié par rapport aux

autres articles promus. En fonction de votre cas d'utilisation ou de votre recette, les articles recommandés qui ne font pas partie d'une promotion sont triés en fonction de leur pertinence pour l'utilisateur, de leur popularité ou de leur similitude. S'il n'y a pas assez d'articles répondant aux critères de promotion, le résultat contiendra autant d'articles sponsorisés que possible.

Articles de promotion (AWS CLI)

Le code suivant montre comment promouvoir des articles dans les recommandations à l'aide de la campagne AWS CLI et d'une campagne personnalisée. Pour promouvoir des articles à l'aide d'un recommandeur, remplacez le `campaign-arn` paramètre par un `recommender-arn` et spécifiez l'Amazon Resource Name (ARN) pour le recommandateur. Pour les champs de promotion, spécifiez les éléments suivants :

- nom : Donnez un nom à la promotion. La réponse à la recommandation utilise le nom pour identifier les articles promus.
- pourcent-promoted-items : pourcentage d'articles recommandés auxquels appliquer la promotion. Dans cet exemple, 50 % des articles seront des articles promus.
- FilterArn : Spécifiez le nom de ressource Amazon (ARN) du filtre qui définit les critères de promotion. Pour plus d'informations, consultez Filtres de promotion.
- noms et valeurs des paramètres : si votre expression de filtre comporte des paramètres, indiquez les noms des paramètres (en distinguant majuscules et minuscules) et les valeurs. Par exemple, si votre expression de filtre comporte un `$GENRE` paramètre, indiquez GENRE comme clé et un ou plusieurs genres, tels que Comédie, comme valeur. Séparez les valeurs multiples par une virgule. Lorsque vous utilisez le AWS CLI, pour chaque valeur, vous devez utiliser le `/` caractère pour éviter à la fois les guillemets et le `/` caractère. L'exemple de code suivant montre comment formater les valeurs.

Le code indique comment utiliser à la fois un filtre de demande et un filtre de promotion. Un filtre de promotion s'applique uniquement aux articles promus, tandis qu'un filtre de demande s'applique uniquement aux articles recommandés restants. Pour plus d'informations, consultez Filtres de promotion.

Pour plus d'informations sur les champs supplémentaires, reportez-vous Obtenir des recommandations d'articles (AWS SDKs) aux sections etObtenir un classement personnalisé à l'aide de métadonnées contextuelles.

```
aws personalize-runtime get-recommendations \
```

```
--campaign-arn CampaignArn \
--user-id 1 \
--num-results 10  \
--filter-arn RequestFilterArn \
--filter-values '{
    "RequestFilterParameterName": "\"value\"",
    "RequestFilterParameterName": "\"value1\",\"value2\",\"value3\""
  }' \
--promotions "[{
  \"name\": \"promotionName\",
  \"percentPromotedItems\": 50,
  \"filterArn\": \"PromotionFilterARN\",
  \"filterValues\": {\"PromotionParameterName\":\"\\\"value1, value2\\\"\"}
}]"
```

La liste des éléments recommandés s'affiche. Les articles sponsorisés sont positionnés de manière aléatoire par rapport aux autres articles recommandés, mais dans un ordre trié par rapport aux autres articles promus. En fonction de votre recette, les articles recommandés qui ne font pas partie d'une promotion sont triés en fonction de leur pertinence pour l'utilisateur, de leur popularité ou de leur similitude. S'il n'y a pas assez d'articles répondant aux critères de promotion, le résultat contiendra autant d'articles sponsorisés que possible.

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

Articles de promotion (AWS SDK)

Le code suivant montre comment promouvoir des éléments dans les recommandations à l' AWS SDK pour Python (Boto3) aide du SDK pour Java 2.x et d'une campagne personnalisée. Pour promouvoir des articles à l'aide d'un recommandeur, remplacez le `campaignArn` paramètre par `recommenderArn` et spécifiez l'Amazon Resource Name (ARN) pour le recommandateur. Pour les champs de promotion, spécifiez les éléments suivants :

- nom : Spécifiez le nom de la promotion. La réponse à la recommandation inclut le nom permettant d'identifier les articles promus.
- pourcentage PromotedItems : pourcentage d'articles recommandés auxquels appliquer la promotion.
- PromotionFilterArn : nom de ressource Amazon (ARN) du filtre qui définit les critères de promotion. Pour plus d’informations, consultez Filtres de promotion.
- Tous les noms et valeurs de paramètres : si votre expression de filtre comporte des paramètres, pour chaque paramètre de votre expression de filtre, indiquez le nom du paramètre (en distinguant majuscules et minuscules) et les valeurs. Par exemple, si votre expression de filtre comporte un `$GENRE` paramètre, `"GENRE"` indiquez-le comme clé et un ou plusieurs genres, tels que « \" Comédie "\ », comme valeur. Séparez les valeurs multiples par une virgule. Par exemple : `"\"comedy\",\"drama\",\"horror"\"`.

Le code suivant montre comment utiliser à la fois un filtre de demande et un filtre de promotion. Un filtre de promotion s'applique uniquement aux articles promus, tandis qu'un filtre de demande s'applique uniquement aux articles recommandés restants. Pour plus d’informations, consultez Filtres de promotion.

Pour plus d'informations sur les champs supplémentaires, reportez-vous Obtenir des recommandations d'articles (AWS SDKs) aux sections etObtenir un classement personnalisé à l'aide de métadonnées contextuelles.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
  campaignArn = "CampaignARN",
```

```
  userId = '1',
  numResults = 10,
  filterArn = 'RequestFilterARN',
  filterValues = {
      "RequestFilterParameterName": "\"value1\"",
      "RequestFilterParameterName": "\"value1\",\"value2\",\"value3\""
      ....
  },
  promotions = [{
    "name" : "promotionName",
    "percentPromotedItems" : 50,
    "filterArn": "promotionFilterARN",
    "filterValues": {
      "PromotionParameterName": "\"Value1\",\"Value2\""
      ...
    }
  }]
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
    if ("promotionName" in item):
        print(item['promotionName'])
```

SDK for Java 2.x

```
public static void getRecommendationsWithPromotedItems(PersonalizeRuntimeClient
 personalizeRuntimeClient,
                                        String campaignArn,
                                        String userId,
                                        String requestFilterArn,
                                        String requestParameterName,
                                        String requestParameterValue1,
                                        String requestParameterValue2,
                                        String promotionName,
                                        int percentPromotedItems,
                                        String promotionFilterArn,
                                        String promotionParameterName,
                                        String promotionParameterValue1,
                                        String promotionParameterValue2) {

  try {
```

```
      Map<String, String> promotionFilterValues = new HashMap<>();

      promotionFilterValues.put(promotionParameterName, String.format("\"%1$s\",
\"%2$s\"",
              promotionParameterValue1, promotionParameterValue2));

      Promotion newPromotion = Promotion.builder()
              .name(promotionName)
              .percentPromotedItems(percentPromotedItems)
              .filterArn(promotionFilterArn)
              .filterValues(promotionFilterValues)
              .build();

      List<Promotion> promotionList = new List<>();

      promotionList.add(newPromotion);

      Map<String, String> requestfilterValues = new HashMap<>();

      requestfilterValues.put(requestParameterName, String.format("\"%1$s\",\"%2$s
\"",
              requestParameterValue1, requestParameterValue2));

      GetRecommendationsRequest recommendationsRequest =
 GetRecommendationsRequest.builder()
              .campaignArn(campaignArn)
              .numResults(20)
              .userId(userId)
              .filterArn(requestFilterArn)
              .fitlerValues(requestFilterValues)
              .promotions(promotionList)
              .build();

      GetRecommendationsResponse recommendationsResponse =
 personalizeRuntimeClient.getRecommendations(recommendationsRequest);
      List<PredictedItem> items = recommendationsResponse.itemList();

      for (PredictedItem item: items) {
          System.out.println("Item Id is : "+item.itemId());
          System.out.println("Item score is : "+item.score());
          System.out.println("Promotion name is : "+item.promotionName());
      }
  } catch (PersonalizeRuntimeException e) {
```

```
      System.err.println(e.awsErrorDetails().errorMessage());
      System.exit(1);
  }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { GetRecommendationsCommand, PersonalizeRuntimeClient } from
  "@aws-sdk/client-personalize-runtime";

// create personalizeRuntimeClient.
const personalizeRuntimeClient = new PersonalizeRuntimeClient({
  region: "REGION",
});

// set recommendation request param
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN",  /* required */
  userId: "USER_ID", /* required */
  numResults: 25, /* optional */
  filterArn: "FILTER_ARN", /* provide if you are applying a custom filter */
  filterValues: {
    "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your filter has a placeholder
 parameter */
  },
  promotions: [
    {
      name: "PROMOTION_NAME", /* specify the name of the promotion. The
 recommendation response includes the name to identify promoted items. */
      percentPromotedItems: 50, /* the percentage of recommended items to apply the
 promotion to. */
      filterArn:
        "PROMOTION_FILTER_ARN", /* the Amazon Resource Name (ARN) of the filter that
 defines the promotion criteria. */
      filterValues: {
        "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your promotion filter has a
 placeholder parameter */
      },
    },
  ],
};
```

```
export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
 GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", "\nItems are: ");
    response.itemList.forEach(element => console.log(element.itemId))
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

La liste des éléments recommandés s'affiche. Les articles sponsorisés sont positionnés de manière aléatoire par rapport aux autres articles recommandés, mais dans un ordre trié par rapport aux autres articles promus. En fonction de votre recette, les articles recommandés qui ne font pas partie d'une promotion sont triés en fonction de leur pertinence pour l'utilisateur, de leur popularité ou de leur similitude. S'il n'y a pas assez d'articles répondant aux critères de promotion, le résultat contiendra autant d'articles sponsorisés que possible.

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

Recommandations d'action en temps réel dans Amazon Personalize

Si vous utilisez une recette PERSONALIZED_ACTIONS, vous pouvez obtenir des recommandations d'action à partir de votre campagne en temps réel. Vous pouvez obtenir des recommandations d'action à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Rubriques

- Comment fonctionne la notation des recommandations d'action
- Obtenir des recommandations d'action (console)
- Obtenir des recommandations d'action (AWS CLI)
- Obtenir des recommandations d'action (AWS SDKs)

Comment fonctionne la notation des recommandations d'action

Avec Next-Best-Action cette recette, Amazon Personalize génère des scores pour les actions en fonction de la probabilité que l'utilisateur interagisse avec l'action. Les scores peuvent être compris entre 0 et 1,0. Plus la valeur est proche de 1.0, plus il est probable que l'utilisateur interagisse avec l'action.

Si vous n'avez importé aucune donnée d'interaction, toutes les actions recommandées auront un score de 0,0. Si Amazon Personalize recommande une action dans le cadre de l'exploration, l'article aura un score de 0,0. Amazon Personalize utilise l'exploration pour recommander des actions sans données d'interaction entre les actions. Pour plus d'informations sur l'exploration, consultezExploration.

Obtenir des recommandations d'action (console)

Pour obtenir des recommandations d'action avec la console Amazon Personalize, vous devez fournir les informations relatives à la demande sur la page détaillée de votre campagne personnalisée.

Pour obtenir des recommandations d'action

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne que vous utilisez.

3. Dans le volet de navigation, sous Ressources personnalisées, sélectionnez Campagnes.
4. Choisissez la campagne cible.
5. Sous Résultats de la campagne de test, entrez les détails de votre demande de recommandation.

   Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.
6. Choisissez éventuellement un filtre. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.
7. Choisissez Get recommendations (Obtenir des recommandations). Un tableau contenant les 5 principales actions recommandées par l'utilisateur s'affiche.

Obtenir des recommandations d'action (AWS CLI)

Utilisez le code suivant pour obtenir des recommandations d'action dans le cadre d'une campagne. Spécifiez l'ID de l'utilisateur pour lequel vous souhaitez obtenir des recommandations et le nom de ressource Amazon (ARN) de votre campagne.

Pour modifier le nombre d'actions recommandées, modifiez la valeur de`numResults`. La valeur par défaut est de 5 actions. Le maximum est de 100 actions.

Pour filtrer les recommandations d'actions selon des critères personnalisés, vous pouvez créer un filtre et l'appliquer à l'`get-action-recommendations`opération. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.

```
aws personalize-runtime get-action-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
```

```
--num-results 10
```

Obtenir des recommandations d'action (AWS SDKs)

Le code suivant montre comment obtenir des recommandations Amazon Personalize pour un utilisateur à partir d'une campagne. Spécifiez l'ID de l'utilisateur pour lequel vous souhaitez obtenir des recommandations, ainsi que le nom de ressource Amazon (ARN) de votre campagne.

Pour modifier le nombre d'actions recommandées, modifiez la valeur de`numResults`. La valeur par défaut est de 5 actions. Le maximum est de 100 actions.

Pour filtrer les recommandations d'actions selon des critères personnalisés, vous pouvez créer un filtre et l'appliquer à la demande GetActionRecommendations d'API. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Si vous avez enregistré des événements pour un utilisateur avant qu'il ne se connecte (utilisateur anonyme), vous pouvez obtenir des recommandations pour cet utilisateur en fournissant les informations `sessionId` issues de ces événements comme s'il s'agissait du sien`userId`. Pour plus d'informations sur l'enregistrement d'événements pour les utilisateurs anonymes, consultezEnregistrement d'événements pour les utilisateurs anonymes.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_action_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)

print("Recommended actions")
for item in response['actionList']:
    print (item['actionId'])
```

Obtenir un classement personnalisé (ressources personnalisées)

Un classement personnalisé est une liste d'éléments recommandés ayant fait l'objet d'un nouveau classement pour un utilisateur spécifique. Pour obtenir des classements personnalisés, appelez

l'opération GetPersonalizedRanking API ou obtenez des recommandations à partir d'une campagne dans la console.

> **Note**
>
> La solution utilisée pour la campagne doit avoir été créée à l'aide d'une recette de type PERSONALIZED_RANKING. Pour de plus amples informations, veuillez consulter Choisir une recette.

Rubriques

- Comment fonctionne le classement personnalisé
- Obtenir un classement personnalisé (console)
- Obtenir un classement personnalisé (AWS CLI)
- Obtenir un classement personnalisé (AWS SDKs)
- Exemple de carnet de notes à classement personnalisé

Comment fonctionne le classement personnalisé

À l'instar des scores renvoyés par l'`GetRecommendations`opération pour les solutions créées avec les recettes User-Personalization-v 2 et de personnalisation par l'utilisateur, la somme des `GetPersonalizedRanking` scores est égale à 1, mais seuls les éléments d'entrée reçoivent des scores et les scores de recommandation ont tendance à être plus élevés. Si un objet n'était pas présent lors du dernier entraînement, il reçoit un score de 0.

Mathématiquement, la fonction de notation pour GetPersonalizedRanking est identique à`GetRecommendations`, sauf qu'elle ne prend en compte que les éléments d'entrée. Cela signifie que des scores plus proches de 1 deviennent plus probables, car les choix pour diviser le score sont plus réduits :

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_{j \in \text{input}} \exp(\bar{w}_u^\top w_j)}$$

Obtenir un classement personnalisé (console)

Pour obtenir un classement personnalisé pour un utilisateur depuis la console Amazon Personalize, choisissez la campagne que vous utilisez, puis fournissez son ID utilisateur, spécifiez la liste

des éléments que vous souhaitez classer pour l'utilisateur, choisissez éventuellement un filtre et fournissez éventuellement des données contextuelles.

Pour obtenir un classement personnalisé pour un utilisateur

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne que vous utilisez.
3. Dans le volet de navigation, choisissez Campaigns (Campagnes).
4. Sur la page Campagnes, choisissez la campagne cible.
5. Sous Résultats de la campagne de test, entrez le nom d'utilisateur de l'utilisateur pour lequel vous souhaitez obtenir des recommandations.
6. Pour Article IDs, entrez la liste des éléments à classer pour l'utilisateur.
7. Choisissez éventuellement un filtre. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.
8. Si vous avez activé les métadonnées dans les recommandations pour votre campagne, pour les colonnes du jeu de données Items, choisissez les colonnes de métadonnées que vous souhaitez inclure dans les résultats des recommandations. Pour plus d'informations sur l'activation des métadonnées, consultezMétadonnées des éléments dans les recommandations.
9. Si votre campagne utilise des métadonnées contextuelles (pour les exigences, voirAméliorer la pertinence des recommandations grâce aux métadonnées contextuelles), fournissez éventuellement des données contextuelles.

   Pour chaque contexte, pour la clé, entrez le champ de métadonnées, et pour la valeur, entrez les données du contexte.
10. Choisissez Obtenir un classement personnalisé des articles. Un tableau contenant les éléments classés par ordre d'intérêt prévu pour l'utilisateur apparaît.

Obtenir un classement personnalisé (AWS CLI)

Les exemples de code suivants montrent les différentes variantes de la manière d'obtenir un classement personnalisé avec le AWS CLI.

Rubriques

- Obtenir un classement personnalisé

- Inclure les métadonnées des articles dans un classement personnalisé

Obtenir un classement personnalisé

Utilisez la `get-personalized-ranking` commande suivante pour obtenir un classement personnalisé avec AWS CLI. Spécifiez le nom de ressource Amazon (ARN) pour votre campagne, le nom d'utilisateur de l'utilisateur et fournissez une liste des articles IDs à classer pour l'utilisateur (chacun séparé par un espace). Les éléments à classer doivent figurer dans les données que vous avez utilisées pour entraîner la version de la solution. La liste des recommandations classées s'affiche. Amazon Personalize considère que le premier élément de la liste présente le plus d'intérêt pour l'utilisateur.

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
--input-list 3 4 10 8 12 7
```

Inclure les métadonnées des articles dans un classement personnalisé

Si vous avez activé les métadonnées dans les recommandations pour votre campagne, vous pouvez spécifier les colonnes de métadonnées du jeu de données Items à inclure dans la réponse. Pour plus d'informations sur l'activation des métadonnées, consultezMétadonnées des éléments dans les recommandations.

L'exemple de code suivant montre comment spécifier les colonnes de métadonnées dans le cadre de votre demande de classement personnalisé.

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
--input-list 3 4 10 8 12 7
--metadata-columns "{\"ITEMS\": [\"\"columnNameA\"\",\"\"columnNameB\"\"]}"
```

Obtenir un classement personnalisé (AWS SDKs)

Les exemples de code suivants montrent les différentes variantes de la manière d'obtenir un classement personnalisé avec le AWS SDKs.

Rubriques

- Obtenir un classement personnalisé
- Inclure les métadonnées des articles dans un classement personnalisé
- Obtenir un classement personnalisé à l'aide de métadonnées contextuelles

Obtenir un classement personnalisé

Le code suivant montre comment obtenir un classement personnalisé pour un utilisateur. Spécifiez l'identifiant de l'utilisateur et une liste d'éléments IDs à classer pour l'utilisateur. L'élément IDs doit se trouver dans les données que vous avez utilisées pour entraîner la version de la solution. Une liste de recommandations classées est renvoyée. Amazon Personalize considère que le premier élément de la liste présente le plus d'intérêt pour l'utilisateur.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = ['ItemID1','ItemID2']
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])
```

SDK for Java 2.x

```
public static List<PredictedItem> getRankedRecs(PersonalizeRuntimeClient
 personalizeRuntimeClient,
                                                 String campaignArn,
                                                 String userId,
                                                 ArrayList<String> items) {

    try {
        GetPersonalizedRankingRequest rankingRecommendationsRequest =
 GetPersonalizedRankingRequest.builder()
                .campaignArn(campaignArn)
                .userId(userId)
```

```
                .inputList(items)
                .build();

        GetPersonalizedRankingResponse recommendationsResponse =

 personalizeRuntimeClient.getPersonalizedRanking(rankingRecommendationsRequest);
        List<PredictedItem> rankedItems =
 recommendationsResponse.personalizedRanking();
        int rank = 1;
        for (PredictedItem item : rankedItems) {
            System.out.println("Item ranked at position " + rank + " details");
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
            System.out.println("-------------------------------------------");
            rank++;
        }
        return rankedItems;
    } catch (PersonalizeRuntimeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return null;
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetPersonalizedRankingCommand } from "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set the ranking request parameters.
export const getPersonalizedRankingParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  inputList: ["ITEM_ID_1", "ITEM_ID_2", "ITEM_ID_3", "ITEM_ID_4"],
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
```

```
      new GetPersonalizedRankingCommand(getPersonalizedRankingParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Inclure les métadonnées des articles dans un classement personnalisé

Si vous avez activé les métadonnées dans les recommandations pour votre campagne, vous pouvez spécifier les colonnes de métadonnées du jeu de données Items à inclure dans la réponse. Pour plus d'informations sur l'activation des métadonnées, consultezMétadonnées des éléments dans les recommandations.

L'exemple de code suivant montre comment spécifier les colonnes de métadonnées dans le cadre de votre demande de classement personnalisé.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = ['ItemID1','ItemID2'],
    metadataColumns = {
      "ITEMS": ['columnNameA','columnNameB']
    }
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])
    print (item['metadata'])
```

Obtenir un classement personnalisé à l'aide de métadonnées contextuelles

Utilisez le code suivant pour obtenir un classement personnalisé basé sur les métadonnées contextuelles. Pour `context` chaque paire clé-valeur, indiquez le champ de métadonnées comme clé et les données contextuelles comme valeur. Dans l'exemple de code suivant, la clé est `DEVICE` et la valeur est`mobile phone`. Remplacez ces valeurs et le `Campaign ARN` et `User ID` par les vôtres. Passez également `inputList` à une liste des éléments IDs figurant dans les données que vous avez utilisées pour entraîner la solution. Amazon Personalize considère que le premier élément de la liste présente le plus d'intérêt pour l'utilisateur.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign ARN",
    userId = "User ID",
    inputList = ['ItemID1', 'ItemID2'],
    context = {
      'DEVICE': 'mobile phone'
    }
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
  print(item['itemId'])
```

Exemple de carnet de notes à classement personnalisé

Pour un exemple de bloc-notes Jupyter expliquant comment utiliser la recette de classement personnalisé, voir l'exemple de Personalize Ranking.

Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles

Pour améliorer la pertinence des recommandations, incluez des métadonnées contextuelles relatives à un utilisateur, telles que son type d'appareil ou l'heure de la journée, lorsque vous recevez des recommandations d'articles ou obtenez un classement personnalisé.

Pour utiliser des métadonnées contextuelles, le schéma du jeu de données d'interactions entre éléments doit comporter des champs de métadonnées pour les données contextuelles. Par exemple, un champ DEVICE (voirCréation de fichiers JSON pour les schémas Amazon Personalize). Lorsque vous considérez les champs à utiliser pour les métadonnées contextuelles, sélectionnez les champs dans lesquels les valeurs sont disponibles pour les données historiques dans le jeu de données sur les interactions entre articles, et lorsque vous recevez des recommandations d'articles ou obtenez un classement personnalisé des éléments.

Pour les groupes de jeux de données de domaine, les cas d'utilisation des recommandations suivants peuvent utiliser des métadonnées contextuelles :

- Recommandé pour vous(domaine ECOMMERCE)
- Les meilleurs choix pour vous(domaine VIDEO_ON_DEMAND)

Pour les ressources personnalisées, les recettes qui utilisent des métadonnées contextuelles sont les suivantes :

- Personnalisation de l'utilisateur-v2 et Personnalisation par l'utilisateur
- Ranking-V2 personnalisé et Personalized-Ranking

Pour plus d'informations sur les informations contextuelles, consultez le billet de blog sur le AWS Machine Learning suivant : Augmenter la pertinence de vos recommandations Amazon Personalize en tirant parti des informations contextuelles.

Vous pouvez obtenir des recommandations avec des métadonnées contextuelles à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI), ou AWSSDKs.

Obtenir des recommandations à l'aide de métadonnées contextuelles (SDK AWS Python)

Pour améliorer la pertinence des recommandations, incluez des métadonnées contextuelles relatives à un utilisateur, telles que son type d'appareil ou l'heure de la journée, lorsque vous recevez des recommandations d'articles ou obtenez un classement personnalisé.

Utilisez le code suivant pour obtenir une recommandation basée sur les métadonnées contextuelles. Pour `context` chaque paire clé-valeur, indiquez le champ de métadonnées comme clé et les données contextuelles comme valeur. Dans l'exemple de code suivant, la clé est `DEVICE` et la valeur

estmobile phone. Remplacez ces valeurs et le Campaign ARN et User ID par les vôtres. Si vous avez créé un recommandeur, remplacez-le campaignArn parrecommenderArn. La liste des éléments recommandés pour l'utilisateur s'affiche.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    context = {
      'DEVICE': 'mobile phone'
    }
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Obtenir des recommandations d'articles par lots grâce à des ressources personnalisées

Grâce aux ressources personnalisées, vous pouvez obtenir des recommandations d'articles grâce à un flux de lots asynchrone. Par exemple, vous pouvez obtenir des recommandations de produits pour tous les utilisateurs d'une liste d'e-mails ou des item-to-itemsimilitudes dans un inventaire.

Pour obtenir des recommandations par lots pour des articles, vous devez utiliser une tâche d'inférence par lots. Une tâche d'inférence par lots est un outil qui importe vos données d'entrée par lots depuis un compartiment Amazon S3, utilise la version de votre solution personnalisée pour générer des recommandations d'articles, puis exporte les recommandations d'articles vers un compartiment Amazon S3. Selon la recette, vos données d'entrée sont une liste d'utilisateurs, ou d'éléments, ou une liste d'utilisateurs possédant chacun une collection d'éléments.

Si votre solution utilise la recette Articles similaires et que vous disposez d'un jeu de données d'articles contenant des données textuelles et des données de titre d'article, vous pouvez générer des recommandations par lots avec des thèmes pour chaque groupe d'articles. Pour de plus amples informations, veuillez consulter Recommandations par lots avec des thèmes issus du générateur de contenu.

Une fois que vous avez créé une version de solution personnalisée, l'influence des nouvelles données sur les recommandations relatives aux articles par lots dépend de leur type, de la méthode d'importation et de la recette personnalisée que vous utilisez. Pour plus d'informations sur la manière dont les nouvelles données influencent les recommandations par lots, voirComment les nouvelles données influencent les recommandations par lots (ressources personnalisées).

Rubriques

- Flux de travail par lots
- Directives et exigences
- Notation du flux de travail par lots
- Recommandations par lots avec des thèmes issus du générateur de contenu
- Préparation des données d'entrée pour les recommandations par lots
- Création d'une tâche d'inférence par lots
- Exemples de résultats de tâches d'inférence par lots

Flux de travail par lots

Le flux de travail par lots est le suivant :

1. Préparez et téléchargez vos données d'entrée au format JSON dans un compartiment Amazon S3. Le format de vos données d'entrée dépend de la recette que vous utilisez. Consultez Préparation des données d'entrée pour les recommandations par lots.
2. Créez un emplacement distinct pour vos données de sortie, qu'il s'agisse d'un dossier ou d'un autre compartiment Amazon S3.
3. Créez une tâche d'inférence par lots. Consultez Création d'une tâche d'inférence par lots.
4. Lorsque l'inférence par lots est terminée, récupérez les recommandations relatives aux articles depuis votre emplacement de sortie dans Amazon S3.

Directives et exigences

Voici les directives et les exigences relatives à l'obtention de recommandations par lots :

- Votre rôle de service Amazon Personalize IAM doit être autorisé à lire et à ajouter des fichiers à vos compartiments Amazon S3. Pour plus d'informations sur l'octroi d'autorisations,

consultezPolitique de rôle de service pour les flux de travail par lots. Pour plus d'informations sur les autorisations relatives aux compartiments, consultez les exemples de politiques utilisateur dans le manuel Amazon Simple Storage Service Developer Guide. Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

- Vous devez créer une solution personnalisée et une version de solution avant de créer une tâche d'inférence par lots. Cependant, il n'est pas nécessaire de créer une campagne Amazon Personalize. Si vous avez créé un groupe de jeux de données de domaine, vous pouvez toujours créer des ressources personnalisées.
- Pour générer des thèmes avec des recommandations, vous devez utiliser la recette Similar-Items. Et vous devez disposer d'un jeu de données Items contenant des données textuelles et des données de titre d'article. Pour plus d'informations sur les recommandations thématiques, voirRecommandations par lots avec des thèmes issus du générateur de contenu.
- Vos données d'entrée doivent être formatées comme décrit dansPréparation des données d'entrée pour les segments d'utilisateurs.
- Vous ne pouvez pas obtenir de recommandations par lots avec le Trending-Now ou les recettes. Next-Best-Action
- Si vous utilisez un filtre avec des paramètres d'espace réservé, vous devez inclure les valeurs des paramètres dans les données d'entrée d'un `filterValues` objet. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.
- Nous vous recommandons d'utiliser un emplacement différent pour vos données de sortie (un dossier ou un compartiment Amazon S3 différent) de celui de vos données d'entrée.
- Les recommandations par lots peuvent ne pas être exactement les mêmes que les recommandations en temps réel. Cela est dû au fait que les tâches d'inférence par lots prennent plus de temps à réaliser et ne prennent en compte les données disponibles que 15 minutes avant le début de la tâche.

Notation du flux de travail par lots

Les recommandations relatives aux lots incluent les scores suivants :

- À l'aide des recettes de personnalisation des utilisateurs et de classement personnalisé, Amazon Personalize calcule les scores de recommandation de travail par inférence par lots, comme décrit dans et. Comment fonctionne la notation des recommandations (ressources personnalisées)

Comment fonctionne le classement personnalisé Vous pouvez consulter les scores dans le fichier JSON de sortie de la tâche d'inférence par lots.

- Avec la recette Similar-Items, si vous recevez des recommandations par lots thématiques, Amazon Personalize classe chaque ensemble d'articles connexes en fonction de la pertinence du thème pour chaque article. Chaque item inclut un score de 0 à 1. Plus le score est élevé, plus l'objet est étroitement lié au thème. Pour plus d'informations sur les recommandations avec des thèmes, consultezRecommandations par lots avec des thèmes issus du générateur de contenu.

Recommandations par lots avec des thèmes issus du générateur de contenu

> **Important**
>
> Lorsque vous recevez des recommandations par lots avec des thèmes, vous encourez des coûts supplémentaires. Pour plus d'informations, consultez les tarifs d'Amazon Personalize.

Si vous utilisez leRecette d'articles similaires, Amazon Personalize Content Generator peut ajouter des thèmes descriptifs aux recommandations par lots. Le générateur de contenu est une fonctionnalité d'intelligence artificielle générative (IA générative) gérée par Amazon Personalize.

Lorsque vous recevez des recommandations par lots avec des thèmes, Amazon Personalize Content Generator ajoute un thème descriptif pour chaque ensemble d'articles similaires. Le thème est basé sur la description de l'article et les données du nom de l'article dans votre jeu de données d'articles. Amazon Personalize inclut les thèmes dans le résultat de la tâche d'inférence par lots. Vous pouvez utiliser les thèmes pour rendre le texte de votre application ou de vos messages marketing plus convaincants.

Par exemple, si vous recevez des recommandations d'articles connexes pour un plat pour le petit déjeuner, Amazon Personalize peut générer un thème tel que Rise and shine ou Morning Essentials. Vous pouvez utiliser le thème pour remplacer un titre de carrousel générique, comme Frequently bought together. Vous pouvez également intégrer le thème dans un e-mail promotionnel ou une campagne marketing pour de nouvelles options de menu.

AWS ne surveille pas les thèmes depuis Content Generator. Pour confirmer la qualité du thème, vous pouvez utiliser les scores produits pour chaque élément recommandé. Pour plus d’informations, consultez Classement et notation pour les recommandations par lots avec des thèmes.

Rubriques

- Régions prises en charge
- Directives et exigences
- Classement et notation pour les recommandations par lots avec des thèmes
- Génération de recommandations par lots avec des thèmes

Régions prises en charge

Amazon Personalize Content Generator n'est disponible que dans les AWS régions suivantes :

- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Asie Pacifique (Tokyo)

Directives et exigences

Voici les directives et les exigences relatives à la génération de recommandations avec des thèmes :

- Votre fichier d'entrée peut contenir jusqu'à 100 éléments. Pour plus d'informations sur les données d'entrée pour les recommandations par lots, voirPréparation des données d'entrée pour les recommandations par lots.
- Votre solution doit utiliser leRecette d'articles similaires.
- Vous devez disposer d'un jeu de données Items contenant les données suivantes. Ces données peuvent aider à générer des thèmes plus pertinents.
  - Il doit comporter un champ textuel, tel qu'un champ DESCRIPTION. Pour plus d'informations sur les données textuelles, consultezMétadonnées de texte non structurées.
  - Il doit comporter une colonne de chaîne contenant les données relatives au nom de l'élément, par exemple un champ TITLE.

  Si votre jeu de données Items ne contient pas ces données, vous pouvez les ajouter. Pour plus d'informations sur la mise à jour des données existantes, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Classement et notation pour les recommandations par lots avec des thèmes

Lorsque vous recevez des recommandations par lots avec des thèmes, Amazon Personalize classe chaque ensemble d'articles en fonction de la pertinence du thème pour chaque article. Chaque élément inclut un score compris approximativement entre -0,1 et 0,6. Plus le score est élevé, plus l'objet est étroitement lié au thème. Vous pouvez utiliser les scores pour définir un seuil afin de n'afficher que les éléments étroitement liés au thème.

Par exemple, Amazon Personalize peut renvoyer le thème suivant : bonbons durs (score 0,19884521)`For your sweet tooth`, chocolat (score 0,17664525), pomme (score 0,08994528), popsicle (score 0,14294521), patate douce (score 0,07794527) et carotte (score 0,04994523). Dans votre application, vous pouvez ajouter une règle pour inclure uniquement les éléments ayant un score égal `.10` ou supérieur, en éliminant les fruits et légumes.

L'exemple suivant montre le format de sortie d'une tâche d'inférence par lots qui génère des recommandations de films avec des thèmes.


```
{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
 with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135314
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
 1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.138913,-
...
```


Génération de recommandations par lots avec des thèmes

Pour générer des recommandations par lots avec des thèmes, vous devez exécuter le flux de travail par lots comme décrit dansFlux de travail par lots. Vous préparez vos données d'entrée de la même manière que pour une `RELATED_ITEMS` recette. Pour obtenir un exemple, consultez Recettes RELATED_ITEMS.

Lorsque vous créez la tâche d'inférence par lots, vous activez la génération de thèmes et vous spécifiez la colonne de titre de l'article de votre jeu de données Items.

- Pour plus d'informations sur l'utilisation de la console Amazon Personalize pour créer une tâche d'inférence par lots qui génère des thèmes, consultezCréation d'une tâche d'inférence par lots.

- Pour un exemple de code montrant comment utiliser le pour AWS SDK pour Python (Boto3) créer une tâche d'inférence par lots générant des thèmes, consultezCréation d'une tâche d'inférence par lots qui génère des thèmes.

Préparation des données d'entrée pour les recommandations par lots

Une tâche d'inférence par lots importe les données JSON saisies par lots depuis un compartiment Amazon S3, utilise la version de votre solution personnalisée pour générer des recommandations, puis exporte les recommandations d'articles vers un compartiment Amazon S3. Avant de pouvoir obtenir des recommandations par lots, vous devez préparer et télécharger votre fichier JSON dans un compartiment Amazon S3. Nous vous recommandons de créer un dossier de sortie dans votre compartiment Amazon S3 ou d'utiliser un compartiment de sortie Amazon S3 distinct. Vous pouvez ensuite exécuter plusieurs tâches d'inférence par lots en utilisant le même emplacement de données d'entrée.

Si vous utilisez un filtre avec des paramètres d'espace réservé, par exemple`$GENRE`, vous devez fournir les valeurs des paramètres d'un `filterValues` objet dans votre code JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.

Pour préparer et importer des données

1. Formatez les données de saisie par lots en fonction de votre recette. Vous ne pouvez pas obtenir de recommandations par lots avec la recette Trending-Now.

   - Pour les recettes USER_PERSONALIZATION et Popularity-Count, vos données d'entrée sont un fichier JSON contenant une liste d'identifiants utilisateur
   - Pour les recettes RELATED_ITEMS, vos données d'entrée sont une liste d'ItemID
   - Pour les recettes PERSONALIZED_RANKING, vos données d'entrée sont une liste d'identifiants d'utilisateur, chacun associé à une collection d'identifiants ItemID

   Séparez chaque ligne par une nouvelle ligne. Pour des exemples de données d'entrée, voirExemples JSON d'entrée et de sortie de tâches d'inférence par lots.
2. Téléchargez votre code JSON d'entrée dans un dossier d'entrée de votre compartiment Amazon S3. Pour plus d'informations, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service

3. Créez un emplacement distinct pour vos données de sortie, qu'il s'agisse d'un dossier ou d'un autre compartiment Amazon S3. En créant un emplacement distinct pour le JSON de sortie, vous pouvez exécuter plusieurs tâches d'inférence par lots avec le même emplacement de données d'entrée.
4. Créez une tâche d'inférence par lots. Amazon Personalize transmet les recommandations de la version de votre solution à l'emplacement de vos données de sortie.

Exemples JSON d'entrée et de sortie de tâches d'inférence par lots

La façon dont vous formatez vos données d'entrée et la recette que vous utilisez. Si vous utilisez un filtre avec des paramètres d'espace réservé, par exemple`$GENRE`, vous devez fournir les valeurs des paramètres d'un `filterValues` objet dans votre code JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.

Les sections suivantes répertorient des exemples d'entrée et de sortie JSON correctement formatés pour les tâches d'inférence par lots. Vous ne pouvez pas obtenir de recommandations par lots avec la recette Trending-Now.

Rubriques

- Recettes USER_PERSONALIZATION (PERSONNALISATION DE L'UTILISATEUR)
- Recettes POPULAR_ITEMS (Popularity-Count uniquement)
- Recettes PERSONALIZED_RANKING
- Recettes RELATED_ITEMS

Recettes USER_PERSONALIZATION (PERSONNALISATION DE L'UTILISATEUR)

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour les recettes USER_PERSONALIZATION. Si vous utilisez User-Personalization-v 2, chaque élément recommandé inclut une liste des raisons pour lesquelles l'élément a été inclus dans les recommandations. Cette liste peut être vide. Pour plus d'informations sur les raisons possibles, voirMotifs de recommandation avec User-Personalization-v 2.

Input

Séparez chacune `userId` par une nouvelle ligne comme suit.

```
{"userId": "4638"}
```

```
{"userId": "663"}
{"userId": "3384"}
...
```

Output

```
{"input":{"userId":"4638"},"output":{"recommendedItems":
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","6237
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.
{"input":{"userId":"663"},"output":{"recommendedItems":
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104"
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.
{"input":{"userId":"3384"},"output":{"recommendedItems":
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035"
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.
...
```

Recettes POPULAR_ITEMS (Popularity-Count uniquement)

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour la recette Popularity-Count. Vous ne pouvez pas obtenir de recommandations par lots avec la recette Trending-Now.

Input

Séparez chacune `userId` par une nouvelle ligne comme suit.

```
{"userId": "12"}
{"userId": "105"}
{"userId": "41"}
...
```

Output

```
{"input": {"userId": "12"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "105"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "41"}, "output": {"recommendedItems": ["105", "106", "441"]}}
...
```

Recettes PERSONALIZED_RANKING

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour les recettes PERSONALIZED_RANKING.

Input

Séparez chacune `userId` et la liste des `itemIds` à classer par une nouvelle ligne comme suit.

```
{"userId": "891", "itemList": ["27", "886", "101"]}
{"userId": "445", "itemList": ["527", "55", "901"]}
{"userId": "71", "itemList": ["27", "351", "101"]}
...
```

Output

```
{"input":{"userId":"891","itemList":["27","886","101"]},"output":
{"recommendedItems":["27","101","886"],"scores":[0.48421,0.28133,0.23446]}}
{"input":{"userId":"445","itemList":["527","55","901"]},"output":
{"recommendedItems":["901","527","55"],"scores":[0.46972,0.31011,0.22017]}}
{"input":{"userId":"71","itemList":["29","351","199"]},"output":{"recommendedItems":
["351","29","199"],"scores":[0.68937,0.24829,0.06232]}}
...
```

Recettes RELATED_ITEMS

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour les recettes RELATED_ITEMS.

Input

Séparez chacune `itemId` par une nouvelle ligne comme suit.

```
{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...
```

Output

```
{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
```

```
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}
...
```

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour la recette Similar-Items avec thèmes.

Input

Séparez chacune `itemId` par une nouvelle ligne comme suit.

```
{"itemId": "40"}
{"itemId": "43"}
...
```

Output

```
{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
 with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
 1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.13891
...
```

Création d'une tâche d'inférence par lots

Créez une tâche d'inférence par lots pour obtenir des recommandations d'articles par lots pour les utilisateurs en fonction des données d'entrée d'Amazon S3. Les données d'entrée peuvent être une liste d'utilisateurs ou d'éléments (ou les deux) au format JSON. Vous pouvez créer une tâche d'inférence par lots à l'aide de la console Amazon Personalize, du AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Lorsque vous créez une tâche d'inférence par lots, vous spécifiez les chemins Amazon S3 vers vos emplacements d'entrée et de sortie. Amazon S3 est basé sur un préfixe. Si vous fournissez un préfixe pour l'emplacement des données d'entrée, Amazon Personalize utilise tous les fichiers correspondant à ce préfixe comme données d'entrée. Par exemple, si vous fournissez `s3://`

`amzn-s3-demo-bucket/folderName` un dossier dont le chemin est défini par le nom de compartiments`3://amzn-s3-demo-bucket/folderName_test`, Amazon Personalize utilise tous les fichiers des deux dossiers comme données d'entrée. Pour utiliser uniquement les fichiers d'un dossier spécifique comme données d'entrée, terminez le chemin Amazon S3 par un délimiteur de préfixe, tel que `/` : `s3://amzn-s3-demo-bucket/folderName/` Pour plus d'informations sur la façon dont Amazon S3 organise les objets, consultez Organisation, listage et utilisation de vos objets.

Pour plus d'informations sur le flux de travail par lots dans Amazon Personalize, notamment sur les exigences en matière d'autorisations, la notation des recommandations, ainsi que sur la préparation et l'importation des données d'entrée, consultezObtenir des recommandations d'articles par lots grâce à des ressources personnalisées.

Rubriques

- Création d'une tâche d'inférence par lots (console)
- Création d'une tâche d'inférence par lots ()AWS CLI
- Création d'une tâche d'inférence par lots ()AWS SDKs

Création d'une tâche d'inférence par lots (console)

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche d'inférence par lots. Cette procédure suppose que vous avez déjà créé une solution et une version de solution (modèle entraîné).

Pour créer une tâche d'inférence par lots (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sous Ressources personnalisées, choisissez Batch inference jobs.
4. Choisissez Create batch inference job (Créer une tâche d'inférence par lots).
5. Choisissez le type de tâche d'inférence par lots.

    - Pour générer des recommandations d'articles sans thèmes, choisissez Recommandations d'articles.
    - Si vous utilisez la recette Similar-Items et que vous souhaitez ajouter des thèmes descriptifs à des groupes d'articles similaires, choisissez Recommandations thématiques avec le générateur de contenu. Pour générer des thèmes, vous devez disposer d'un jeu de données

Items contenant des données de nom d'élément et des données textuelles. Pour de plus amples informations, veuillez consulter Recommandations par lots avec des thèmes issus du générateur de contenu.

6. Dans Batch inference job details (Détails de la tâche d'inférence par lots), dans Batch inference job name (Nom de la tâche d'inférence par lots), spécifiez un nom pour votre tâche d'inférence par lots.
7. Pour Solution, choisissez la solution, puis choisissez l'ID de version de solution que vous souhaitez utiliser pour générer les recommandations.
8. Pour Nombre de résultats, spécifiez éventuellement le nombre de recommandations pour chaque ligne de données d'entrée. La valeur par défaut est 25.
9. Si votre traitement par lots génère des recommandations avec des thèmes, dans Détails des recommandations thématiques, choisissez la colonne contenant les noms ou les titres des éléments de votre jeu de données Items. Ces données peuvent aider à générer des thèmes plus pertinents. Pour de plus amples informations, veuillez consulter Recommandations par lots avec des thèmes issus du générateur de contenu.
10. Dans Source d'entrée, spécifiez le chemin Amazon S3 vers votre fichier d'entrée.

    Utilisez la syntaxe suivante : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`**

    Vos données d'entrée doivent être au format adapté à la recette utilisée par votre solution. Pour des exemples de données d'entrée, voirExemples JSON d'entrée et de sortie de tâches d'inférence par lots.
11. Pour la clé de déchiffrement, si vous utilisez votre propre AWS KMS clé pour le chiffrement des compartiments, spécifiez l'Amazon Resource Name (ARN) de votre clé. Amazon Personalize doit être autorisé à utiliser votre clé. Pour plus d'informations sur l'octroi d'autorisations, consultezAutoriser Amazon Personalize à utiliser votre AWS KMS clé.
12. Dans Destination de sortie, spécifiez le chemin d'accès à votre emplacement de sortie. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3).

    Utilisez la syntaxe suivante : **`s3://amzn-s3-demo-bucket/<output folder name>/`**
13. Pour la clé de chiffrement, si vous utilisez votre propre AWS KMS clé pour le chiffrement, spécifiez l'ARN de votre clé. Amazon Personalize doit être autorisé à utiliser votre clé. Pour plus d'informations sur l'octroi d'autorisations, consultezAutoriser Amazon Personalize à utiliser votre AWS KMS clé.

14. Pour le rôle de service IAM, choisissez le rôle de service IAM que vous avez créé pour Amazon Personalize lors de la configuration. Ce rôle doit avoir un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie respectivement.
15. Dans Filtres, choisissez éventuellement un filtre pour appliquer un filtre aux recommandations par lots. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.
16. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
17. Choisissez Create batch inference job (Créer une tâche d'inférence par lots). La création d'une tâche d'inférence par lots démarre et la page Batch inference jobs (Tâches d'inférence par lots) apparaît avec la section Batch inference job detail (Détails de la tâche d'inférence par lots) affichée.

    Lorsque le statut de la tâche d'inférence par lots passe à Active, vous pouvez récupérer le résultat de la tâche dans le compartiment de sortie Amazon S3 désigné. Le nom du fichier de sortie est au format *`input-name`*`.out`.

Création d'une tâche d'inférence par lots ()AWS CLI

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche d'inférence par lots avec l'CreateBatchInferenceJobopération.

Rubriques

- Création d'une tâche d'inférence par lots
- Création d'une tâche d'inférence par lots qui génère des thèmes

Création d'une tâche d'inférence par lots

Vous pouvez utiliser la `create-batch-inference-job` commande pour créer une tâche d'inférence par lots. Spécifiez un nom de tâche, remplacez-le `Solution version ARN` par le nom de ressource Amazon (ARN) de la version de votre solution et remplacez-le par l'`IAM service role ARN`ARN du rôle de service IAM que vous avez créé pour Amazon Personalize lors de la configuration. Ce rôle doit avoir un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie respectivement. Fournissez éventuellement un ARN de filtre pour filtrer les recommandations. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les

valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées).

Remplacez `S3 input path` et `S3 output path` par le chemin Amazon S3 menant à votre fichier d'entrée et à votre emplacement de sortie. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez la syntaxe suivante pour les emplacements d'entrée et de sortie : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** et**`s3://amzn-s3-demo-bucket/<output folder name>/`**.

L'exemple inclut les `itemExplorationConfig` hyperparamètres facultatifs spécifiques à la recette de personnalisation utilisateur : et. `explorationWeight` `explorationItemAgeCutOff` Incluez éventuellement `explorationWeight` des `explorationItemAgeCutOff` valeurs et pour configurer l'exploration. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

```
aws personalize create-batch-inference-job \
--job-name Batch job name \
--solution-version-arn Solution version ARN \
--filter-arn Filter ARN \
--job-input s3DataSource={path=s3://S3 input path} \
--job-output s3DataDestination={path=s3://S3 output path} \
--role-arn IAM service role ARN \
--batch-inference-job-config "{\"itemExplorationConfig\":{\"explorationWeight\":
\"0.3\",\"explorationItemAgeCutOff\":\"30\"}}"
```

Création d'une tâche d'inférence par lots qui génère des thèmes

Pour générer des thèmes pour des éléments similaires, vous devez utiliser la recette Similar-Items et votre jeu de données Items doit comporter un champ textuel et une colonne de données de nom d'élément. Pour plus d'informations sur les recommandations avec des thèmes, consultezRecommandations par lots avec des thèmes issus du générateur de contenu.

Le code suivant crée une tâche d'inférence par lots qui génère des recommandations avec des thèmes. Laissez le `batch-inference-job-mode` set à`THEME_GENERATION`. `COLUMN_NAME`Remplacez-le par le nom de la colonne qui contient les données relatives au nom de l'article.

```
aws personalize create-batch-inference-job \
```

```
--job-name Themed batch job name \
--solution-version-arn Solution version ARN \
--filter-arn Filter ARN \
--job-input s3DataSource={path=s3://S3 input path} \
--job-output s3DataDestination={path=s3://S3 output path} \
--role-arn IAM service role ARN \
--batch-inference-job-mode THEME_GENERATION \
--theme-generation-config "{\"fieldsForThemeGeneration\": {\"itemName\":
\"COLUMN_NAME\"}}"
```

Création d'une tâche d'inférence par lots ()AWS SDKs

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche d'inférence par lots avec l'CreateBatchInferenceJobopération.

Rubriques

- Création d'une tâche d'inférence par lots
- Création d'une tâche d'inférence par lots qui génère des thèmes

Création d'une tâche d'inférence par lots

Vous pouvez utiliser le code suivant pour créer une tâche d'inférence par lots. Spécifiez le nom de la tâche, l'Amazon Resource Name (ARN) de la version de votre solution et l'ARN du rôle de service IAM que vous avez créé pour Amazon Personalize lors de la configuration. Ce rôle doit disposer d'un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie.

Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez la syntaxe suivante pour les emplacements d'entrée et de sortie : **`s3:/amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** et**`s3://amzn-s3-demo-bucket/<output folder name>/`**.

Pour`numResults`, spécifiez le nombre d'articles que vous souhaitez qu'Amazon Personalize prédise pour chaque ligne de données d'entrée. Fournissez éventuellement un ARN de filtre pour filtrer les recommandations. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées).

SDK for Python (Boto3)

L'exemple inclut les `itemExplorationConfig` hyperparamètres facultatifs spécifiques à la recette de personnalisation utilisateur : et. `explorationWeight` `explorationItemAgeCutOff` Incluez éventuellement `explorationWeight` des `explorationItemAgeCutOff` valeurs et pour configurer l'exploration. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM service role ARN",
    filterArn = "Filter ARN",
    batchInferenceJobConfig = {
        # optional USER_PERSONALIZATION recipe hyperparameters
        "itemExplorationConfig": {
            "explorationWeight": "0.3",
            "explorationItemAgeCutOff": "30"
        }
    },
    jobInput =
       {"s3DataSource": {"path": "s3://amzn-s3-demo-bucket/<folder name>/<input JSON
 file name>.json"}},
    jobOutput =
       {"s3DataDestination": {"path": "s3:/amzn-s3-demo-bucket/<output folder
 name>/"}}
)
```

SDK for Java 2.x

L'exemple inclut les `itemExplorationConfig` champs facultatifs spécifiques à la recette de personnalisation utilisateur : `explorationWeight` et. `explorationItemAgeCutOff` Incluez éventuellement `explorationWeight` des `explorationItemAgeCutOff` valeurs et pour configurer l'exploration. Pour de plus amples informations, veuillez consulter Recette de personnalisation par l'utilisateur.

```
public static String createPersonalizeBatchInferenceJob(PersonalizeClient
 personalizeClient,
```

```
                                                    String solutionVersionArn,
                                                    String jobName,
                                                    String filterArn,
                                                    String
s3InputDataSourcePath,
                                                    String
s3DataDestinationPath,
                                                    String roleArn,
                                                    String explorationWeight,
                                                    String
explorationItemAgeCutOff) {

 long waitInMilliseconds = 60 * 1000;
 String status;
 String batchInferenceJobArn;

 try {
     // Set up data input and output parameters.
     S3DataConfig inputSource = S3DataConfig.builder()
             .path(s3InputDataSourcePath)
             .build();
     S3DataConfig outputDestination = S3DataConfig.builder()
             .path(s3DataDestinationPath)
             .build();

     BatchInferenceJobInput jobInput = BatchInferenceJobInput.builder()
             .s3DataSource(inputSource)
             .build();
     BatchInferenceJobOutput jobOutputLocation = BatchInferenceJobOutput.builder()
             .s3DataDestination(outputDestination)
             .build();

     // Optional code to build the User-Personalization specific item exploration
config.
     HashMap<String, String> explorationConfig = new HashMap<>();

     explorationConfig.put("explorationWeight", explorationWeight);
     explorationConfig.put("explorationItemAgeCutOff", explorationItemAgeCutOff);

     BatchInferenceJobConfig jobConfig = BatchInferenceJobConfig.builder()
             .itemExplorationConfig(explorationConfig)
             .build();
     // End optional User-Personalization recipe specific code.
```

```
     CreateBatchInferenceJobRequest createBatchInferenceJobRequest =
CreateBatchInferenceJobRequest.builder()
             .solutionVersionArn(solutionVersionArn)
             .jobInput(jobInput)
             .jobOutput(jobOutputLocation)
             .jobName(jobName)
             .filterArn(filterArn)
             .roleArn(roleArn)
             .batchInferenceJobConfig(jobConfig)   // Optional
             .build();

     batchInferenceJobArn =
personalizeClient.createBatchInferenceJob(createBatchInferenceJobRequest)
             .batchInferenceJobArn();
     DescribeBatchInferenceJobRequest describeBatchInferenceJobRequest =
DescribeBatchInferenceJobRequest.builder()
             .batchInferenceJobArn(batchInferenceJobArn)
             .build();

     long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

     // wait until the batch inference job is complete.
     while (Instant.now().getEpochSecond() < maxTime) {

         BatchInferenceJob batchInferenceJob = personalizeClient
                 .describeBatchInferenceJob(describeBatchInferenceJobRequest)
                 .batchInferenceJob();

         status = batchInferenceJob.status();
         System.out.println("Batch inference job status: " + status);

         if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
             break;
         }
         try {
             Thread.sleep(waitInMilliseconds);
         } catch (InterruptedException e) {
             System.out.println(e.getMessage());
         }
     }
     return batchInferenceJobArn;

 } catch (PersonalizeException e) {
     System.out.println(e.awsErrorDetails().errorMessage());
```

```
  }
  return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchInferenceJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch inference job's parameters.

export const createBatchInferenceJobParam = {
  jobName: "JOB_NAME",
  jobInput: {
    s3DataSource: {
      path: "INPUT_PATH",
    },
  },
  jobOutput: {
    s3DataDestination: {
      path: "OUTPUT_PATH",
    },
  },
  roleArn: "ROLE_ARN",
  solutionVersionArn: "SOLUTION_VERSION_ARN",
  numResults: 20,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateBatchInferenceJobCommand(createBatchInferenceJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
```

```
run();
```

Le traitement de la tâche par lots peut prendre un certain temps. Vous pouvez vérifier l'état d'une tâche en appelant DescribeBatchInferenceJob et en transmettant un `batchRecommendationsJobArn` comme paramètre d'entrée. Vous pouvez également répertorier toutes les tâches d'inférence par lots Amazon Personalize de votre AWS environnement en appelantListBatchInferenceJobs.

Création d'une tâche d'inférence par lots qui génère des thèmes

Pour générer des thèmes pour des éléments similaires, vous devez utiliser la recette Similar-Items et votre jeu de données Items doit comporter un champ textuel et une colonne de données de nom d'élément. Pour plus d'informations sur les recommandations avec des thèmes, consultezRecommandations par lots avec des thèmes issus du générateur de contenu.

Le code suivant crée une tâche d'inférence par lots qui génère des recommandations avec des thèmes. Laissez le `batchInferenceJobMode` set à`"THEME_GENERATION"`. `COLUMNN_NAME`Remplacez-le par le nom de la colonne qui contient les données relatives au nom de l'article.

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM service role ARN",
    filterArn = "Filter ARN",
    batchInferenceJobMode = "THEME_GENERATION",
    themeGenerationConfig = {
      "fieldsForThemeGeneration": {
          "itemName": "COLUMN_NAME"
      }
    },
    jobInput =
       {"s3DataSource": {"path": "s3://amzn-s3-demo-bucket/<folder name>/<input JSON
 file name>.json"}},
    jobOutput =
```

```
        {"s3DataDestination": {"path": "s3://amzn-s3-demo-bucket/<output folder
 name>/"}}
)
```

Exemples de résultats de tâches d'inférence par lots

Lorsque vous créez une tâche d'inférence par lots, la tâche importe vos données d'entrée par lots depuis un compartiment Amazon S3, utilise la version de votre solution pour générer des recommandations d'articles et exporte les recommandations vers un compartiment Amazon S3 au format JSON.

Les sections suivantes répertorient des exemples de fichiers de sortie pour les tâches d'inférence par lots par type de recette. Vous ne pouvez pas obtenir de recommandations par lots avec le Trending-Now ou les recettes. Next-Best-Action

Rubriques

- Recettes USER_PERSONALIZATION (PERSONNALISATION DE L'UTILISATEUR)
- recettes POPULAR_ITEMS
- Recettes PERSONALIZED_RANKING
- Recettes RELATED_ITEMS

Recettes USER_PERSONALIZATION (PERSONNALISATION DE L'UTILISATEUR)

Voici un exemple de fichier JSON de sortie pour une recette USER_PERSONALIZATION.

```
{"input":{"userId":"4638"},"output":{"recommendedItems":
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","62374",
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.004
{"input":{"userId":"663"},"output":{"recommendedItems":
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104","4
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.008
{"input":{"userId":"3384"},"output":{"recommendedItems":
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035","2
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.005
...
```

recettes POPULAR_ITEMS

L'exemple suivant montre le format du fichier JSON de sortie pour la recette Popularity-Count. Vous ne pouvez pas obtenir de recommandations par lots avec la recette Trending-Now.

```
{"input": {"userId": "12"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "105"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "41"}, "output": {"recommendedItems": ["105", "106", "441"]}}
...
```

Recettes PERSONALIZED_RANKING

L'exemple suivant montre le format du fichier JSON de sortie pour une recette PERSONALIZED_RANKING.

```
{"input":{"userId":"891","itemList":["27","886","101"]},"output":{"recommendedItems":
["27","101","886"],"scores":[0.48421,0.28133,0.23446]}}
{"input":{"userId":"445","itemList":["527","55","901"]},"output":{"recommendedItems":
["901","527","55"],"scores":[0.46972,0.31011,0.22017]}}
{"input":{"userId":"71","itemList":["29","351","199"]},"output":{"recommendedItems":
["351","29","199"],"scores":[0.68937,0.24829,0.06232]}}
...
```

Recettes RELATED_ITEMS

L'exemple suivant montre le format du fichier JSON de sortie pour une recette RELATED_ITEMS.

```
{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}
...
```

L'exemple suivant montre le format du fichier JSON de sortie pour la recette Similar-Items avec thèmes. Pour plus d'informations sur les recommandations avec des thèmes, consultezRecommandations par lots avec des thèmes issus du générateur de contenu.

```
{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
 with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135314
```

```
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
 1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.138913,-
...
```


Obtenir des segments d'utilisateurs par lots avec des ressources personnalisées

Pour obtenir des segments d'utilisateurs, vous devez utiliser une tâche de segmentation par lots. Une tâche de segmentation par lots est un outil qui importe vos données d'entrée par lots depuis un compartiment Amazon S3 et utilise la version de votre solution entraînée selon une recette USER_SEGMENTATION pour générer des segments d'utilisateurs pour chaque ligne de données d'entrée.

Selon la recette, les données d'entrée sont une liste d'éléments ou des attributs de métadonnées d'éléments au format JSON. Pour les attributs des articles, vos données d'entrée peuvent inclure des expressions permettant de créer des segments d'utilisateurs basés sur plusieurs attributs de métadonnées. Une tâche de segmentation par lots exporte des segments utilisateur vers un compartiment Amazon S3 en sortie. Chaque segment d'utilisateurs est trié par ordre décroissant en fonction de la probabilité que chaque utilisateur interagisse avec l'élément figurant dans vos données d'entrée.

Lors de la génération de segments d'utilisateurs, Amazon Personalize prend en compte les données des ensembles de données provenant d'importations groupées et individuelles :

- Pour les données en masse, Amazon Personalize génère des segments en utilisant uniquement les données en masse présentes lors de la dernière formation sur la version complète de la solution. Et il utilise uniquement les données en masse que vous avez importées avec le mode d'importation COMPLET (en remplacement des données existantes).
- Pour les données issues d'opérations d'importation de données individuelles, Amazon Personalize génère des segments d'utilisateurs en utilisant les données présentes lors de la dernière formation sur la version complète de la solution. Pour que les nouveaux enregistrements aient un impact sur les segments d'utilisateurs, créez une nouvelle version de la solution, puis créez une tâche de segmentation par lots.

La génération de segments d'utilisateurs fonctionne comme suit :

1. Préparez et téléchargez vos données d'entrée au format JSON dans un compartiment Amazon S3. Le format de vos données d'entrée dépend de la recette que vous utilisez et de la tâche que vous créez. Consultez Préparation des données d'entrée pour les segments d'utilisateurs.
2. Créez un emplacement distinct pour vos données de sortie, qu'il s'agisse d'un dossier différent ou d'un autre compartiment Amazon S3.
3. Créez une tâche de segmentation par lots. Consultez Obtenir des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots.
4. Lorsque le travail de segmentation par lots est terminé, récupérez les segments utilisateur depuis votre emplacement de sortie dans Amazon S3.

Rubriques

- Directives et exigences relatives à l'obtention de segments d'utilisateurs
- Préparation des données d'entrée pour les segments d'utilisateurs
- Obtenir des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots
- Exemples de formats de sortie de tâches de segmentation par lots

Directives et exigences relatives à l'obtention de segments d'utilisateurs

Les directives et les exigences relatives à l'obtention de segments par lots sont les suivantes :

- Vous devez utiliser une recette USER_SEGMENTATION.
- Votre rôle de service Amazon Personalize IAM nécessite une autorisation pour lire et ajouter des fichiers à vos compartiments Amazon S3. Pour plus d'informations sur l'octroi d'autorisations, consultezPolitique de rôle de service pour les flux de travail par lots. Pour plus d'informations sur les autorisations relatives aux compartiments, consultez les exemples de politiques utilisateur dans le manuel Amazon Simple Storage Service Developer Guide.

  Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à utiliser votre AWS KMS clé.
- Vous devez créer une solution personnalisée et une version de solution avant de créer une tâche d'inférence par lots. Cependant, il n'est pas nécessaire de créer une campagne Amazon Personalize. Si vous avez créé un groupe de jeux de données de domaine, vous pouvez toujours créer des ressources personnalisées.

- Vos données d'entrée doivent être formatées comme décrit dansPréparation des données d'entrée pour les segments d'utilisateurs.
- Si vous utilisez la Item-Attribute-Affinity recette, les attributs de vos données d'entrée ne peuvent pas inclure les métadonnées textuelles non structurées des articles, telles qu'une description de produit.
- Si vous utilisez un filtre avec des paramètres d'espace réservé, vous devez inclure les valeurs des paramètres dans les données d'entrée d'un `filterValues` objet. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.
- Nous vous recommandons d'utiliser un emplacement différent pour vos données de sortie (un dossier ou un compartiment Amazon S3 différent) de celui de vos données d'entrée.

Préparation des données d'entrée pour les segments d'utilisateurs

Les tâches de segmentation par lots utilisent une version de solution pour créer des segments d'utilisateurs en fonction des données que vous fournissez dans un fichier JSON d'entrée. Avant de pouvoir obtenir des segments d'utilisateurs, vous devez préparer et télécharger votre fichier JSON dans un compartiment Amazon S3. Nous vous recommandons de créer un dossier de sortie dans votre compartiment Amazon S3 ou d'utiliser un compartiment de sortie Amazon S3 distinct. Vous pouvez ensuite exécuter plusieurs tâches d'inférence par lots en utilisant le même emplacement de données d'entrée.

Si vous utilisez un filtre avec des paramètres d'espace réservé, par exemple`$GENRE`, vous devez fournir les valeurs des paramètres d'un `filterValues` objet dans votre code JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.

Pour préparer et importer des données

1. Formatez les données d'entrée par lots en fonction de la recette utilisée par votre solution. Séparez l'élément de données d'entrée par une nouvelle ligne. Vos données d'entrée sont soit une liste d'ItemID (Item-Affinity) soit des attributs d'article (Item-Attribute-Affinity).

   - Pour les attributs d'article, les données d'entrée peuvent inclure des expressions logiques avec l'`AND`opérateur pour obtenir des utilisateurs pour plusieurs éléments ou attributs par requête. Pour de plus amples informations, veuillez consulter Spécifier les attributs des articles pour la Item-Attribute-Affinity recette.

- Pour les attributs des articles, utilisez le \ caractère pour éviter les caractères spéciaux et les guillemets simples ou doubles dans vos données d'entrée.
- Pour des exemples de données d'entrée pour les deux recettes, voirExemples JSON d'entrée et de sortie de tâches de segmentation par lots.

2. Téléchargez votre code JSON d'entrée dans un dossier d'entrée de votre compartiment Amazon S3. Pour plus d'informations, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service
3. Créez un emplacement distinct pour vos données de sortie, qu'il s'agisse d'un dossier ou d'un autre compartiment Amazon S3. En créant un emplacement distinct pour le JSON de sortie, vous pouvez exécuter plusieurs tâches de segmentation par lots avec le même emplacement de données d'entrée.

Une fois que vous avez préparé vos données d'entrée et que vous les avez téléchargées dans un compartiment Amazon S3, vous êtes prêt à générer des segments d'utilisateurs avec une tâche de segmentation par lots. Pour de plus amples informations, veuillez consulter Obtenir des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots.

Rubriques

- Spécifier les attributs des articles pour la Item-Attribute-Affinity recette
- Exemples JSON d'entrée et de sortie de tâches de segmentation par lots

Spécifier les attributs des articles pour la Item-Attribute-Affinity recette

Si vous utilisez la Item-Attribute-Affinity recette, vos données d'entrée sont une liste d'attributs d'articles. Vous pouvez mélanger différentes colonnes de métadonnées. Par exemple, une ligne peut être une colonne numérique et la suivante une colonne catégorique. Vous ne pouvez pas utiliser les métadonnées textuelles non structurées d'un élément comme attribut d'article.

Les métadonnées de votre élément d'entrée peuvent inclure des expressions logiques avec l'ANDopérateur pour obtenir un segment utilisateur pour plusieurs attributs. Par exemple, une ligne de vos données d'entrée peut être `{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}` ou`{"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.audience = "\teen\""}`.

Lorsque vous combinez deux attributs avec l'ANDopérateur, vous créez un segment d'utilisateurs avec les utilisateurs les plus susceptibles d'interagir avec des éléments dotés des deux attributs en

fonction de l'historique des interactions des utilisateurs. Contrairement aux expressions de filtre (qui utilisent l'`IN`opérateur pour l'égalité des chaînes), les expressions d'entrée de segments par lots ne prennent en charge que le = symbole d'égalité pour la correspondance de chaînes.

Exemples JSON d'entrée et de sortie de tâches de segmentation par lots

Pour une tâche de segmentation par lots, vos données d'entrée doivent être soit une liste d'ItemID (recette Item-Affinity), soit des attributs d'article (Item-Attribute-Affinity). Chaque ligne de données d'entrée est une requête d'inférence distincte. Chaque segment d'utilisateurs est trié par ordre décroissant en fonction de la probabilité que chaque utilisateur interagisse avec les articles de votre inventaire.

Si vous utilisez un filtre avec des paramètres d'espace réservé, par exemple`$GENRE`, vous devez fournir les valeurs des paramètres d'un `filterValues` objet dans votre code JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.

Vous trouverez ci-dessous des exemples d'entrée et de sortie JSON correctement formatés pour les tâches de segmentation par lots organisées par recette.

Affinité entre les objets

Input

Vos données d'entrée peuvent comporter un maximum de 500 éléments. Séparez chacune `itemId` par une nouvelle ligne comme suit.

```
{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...
```

Output

```
{"input": {"itemId": "105"}, "output": {"recommendedUsers": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedUsers": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedUsers": ["2", "442", "435"]}}
...
```

Affinité entre les attributs et les objets

Input

Vos données d'entrée peuvent comporter un maximum de 10 requêtes, chaque requête portant sur un ou plusieurs attributs d'élément non textuels. Séparez chaque attribut ou expression d'attribut par une nouvelle ligne comme suit.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}
{"itemAttributes": "ITEMS.genres = \"Comedy\""}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\""}
...
```

Output

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"",
 "output": {"recommendedUsers": ["25", "78", "108"]}}
{"itemAttributes": "ITEMS.genres = \"Adventure\"", "output": {"recommendedUsers":
 ["87", "31", "129"]}}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"",
 "output": {"recommendedUsers": ["8", "442", "435"]}}
...
```

Obtenir des segments d'utilisateurs à l'aide d'une tâche de segmentation par lots

Si vous avez utilisé une recette USER_SEGMENTATION, vous pouvez créer des tâches de segmentation par lots pour obtenir des segments d'utilisateurs avec la version de votre solution. Chaque segment d'utilisateurs est trié par ordre décroissant en fonction de la probabilité que chaque utilisateur interagisse avec les articles de votre inventaire. Selon la recette, vos données d'entrée doivent être une liste d'éléments (Recette Item-Affinity) ou d'attributs d'éléments (Item-Attribute-Affinity recette) au format JSON. Vous pouvez créer une tâche de segmentation par lots à l'aide de la console Amazon Personalize, du AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Lorsque vous créez une tâche de segmentation par lots, vous spécifiez les chemins Amazon S3 vers vos emplacements d'entrée et de sortie. Amazon S3 est basé sur un préfixe. Si vous fournissez un préfixe pour l'emplacement des données d'entrée, Amazon Personalize utilise tous les fichiers correspondant à ce préfixe comme données d'entrée. Par exemple, si vous fournissez `s3://amzn-s3-demo-bucket/folderName` un dossier dont le chemin est défini par le nom de compartiments`s3://amzn-s3-demo-bucket/folderName_test`, Amazon Personalize utilise tous les fichiers des deux dossiers comme données d'entrée. Pour utiliser uniquement les fichiers d'un

dossier spécifique comme données d'entrée, terminez le chemin Amazon S3 par un délimiteur de préfixe, tel que / : `s3://amzn-s3-demo-bucket/folderName/` Pour plus d'informations sur la façon dont Amazon S3 organise les objets, consultez Organisation, listage et utilisation de vos objets.

Rubriques

- Création d'une tâche de segmentation par lots (console)
- Création d'une tâche de segmentation par lots (AWS CLI)
- Création d'une tâche de segmentation par lots (AWS SDKs)

Création d'une tâche de segmentation par lots (console)

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche de segmentation par lots. Cette procédure suppose que vous avez déjà créé une solution et une version de solution (modèle entraîné) avec une recette USER_SEGMENTATION.

Pour créer une tâche de segmentation par lots (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page du groupe de jeux de données, choisissez votre groupe de jeux de données.
3. Choisissez les tâches de segmentation par lots dans le volet de navigation, puis choisissez Créer une tâche de segmentation par lots.
4. Dans les détails de la tâche de segment par lots, pour Nom de la tâche de segment par lots, spécifiez un nom pour votre tâche de segment par lots.
5. Pour Solution, choisissez la solution, puis choisissez l'ID de version de solution que vous souhaitez utiliser pour générer les recommandations. Vous pouvez créer des tâches de segmentation par lots uniquement si vous avez utilisé une recette USER_SEGMENTATION.
6. Pour Nombre d'utilisateurs, spécifiez éventuellement le nombre d'utilisateurs qu'Amazon Personalize génère pour chaque segment d'utilisateurs. La valeur par défaut est 25. Le maximum est de 5 millions.
7. Pour Source d'entrée, spécifiez le chemin Amazon S3 vers votre fichier d'entrée ou utilisez le bouton Parcourir S3 pour choisir votre compartiment Amazon S3.

   Utilisez la syntaxe suivante : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`**

Vos données d'entrée doivent être au format adapté à la recette utilisée par votre solution. Pour des exemples de données d'entrée, voirExemples JSON d'entrée et de sortie de tâches de segmentation par lots.

8. Pour Destination de sortie, spécifiez le chemin d'accès à votre emplacement de sortie ou utilisez le bouton Parcourir S3 pour choisir votre compartiment Amazon S3. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3).

   Utilisez la syntaxe suivante : **`s3://amzn-s3-demo-bucket/<output folder name>/`**
9. Pour le rôle IAM, choisissez l'une des options suivantes :

   - Choisissez Créer et utiliser un nouveau rôle de service et entrez le nom du rôle de service pour créer un nouveau rôle, ou
   - Si vous avez déjà créé un rôle avec les autorisations appropriées, choisissez Utiliser un rôle de service existant et choisissez le rôle IAM.

   Le rôle que vous utilisez doit avoir un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie respectivement.
10. Pour la configuration du filtre, choisissez éventuellement un filtre pour appliquer un filtre aux segments d'utilisateurs. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Fournir des valeurs de filtre dans votre JSON d'entrée.
11. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
12. Choisissez Créer une tâche segmentée par lots. La création des tâches de segment par lots commence et la page Tâches de segment par lots apparaît avec la section détaillée des tâches de segment par lots affichée.
13. Lorsque le statut de la tâche de segmentation par lots devient Active, vous pouvez récupérer le résultat de la tâche dans le compartiment de sortie Amazon S3 désigné. Le nom du fichier de sortie est au format *`input-name`*`.out`.

Création d'une tâche de segmentation par lots (AWS CLI)

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche de segmentation par lots à l'aide du `create-batch-segment-`

`job` code suivant. Spécifiez un nom de tâche, remplacez-le `Solution version ARN` par le nom de ressource Amazon (ARN) de la version de votre solution et remplacez-le par l'`IAM service role ARN`ARN du rôle de service IAM que vous avez créé pour Amazon Personalize lors de la configuration. Ce rôle doit avoir un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie respectivement. Pour `num-results` spécifier le nombre d'utilisateurs que vous souhaitez qu'Amazon Personalize prédise pour chaque ligne de données d'entrée. La valeur par défaut est 25. Le maximum est de 5 millions. Fournissez éventuellement un `filter-arn` pour filtrer les segments d'utilisateurs. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées).

Remplacez `S3 input path` et `S3 output path` par le chemin Amazon S3 menant à votre fichier d'entrée et à votre emplacement de sortie. Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez la syntaxe suivante pour les emplacements d'entrée et de sortie : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** et**`s3://amzn-s3-demo-bucket/<output folder name>/`**.

```
aws personalize create-batch-segment-job \
                --job-name Job name \
                --solution-version-arn Solution version ARN \
                --num-results The number of predicted users \
                --filter-arn Filter ARN \
                --job-input s3DataSource={path=s3://S3 input path} \
                --job-output s3DataDestination={path=s3://S3 output path} \
                --role-arn IAM service role ARN
{
   "batchSegmentJobArn": "arn:aws:personalize:us-west-2:acct-id:batch-segment-job/
batchSegmentJobName"
}
```

Création d'une tâche de segmentation par lots (AWS SDKs)

Une fois que vous avez terminéPréparation des données d'entrée pour les recommandations par lots, vous êtes prêt à créer une tâche de segmentation par lots avec l'`CreateBatchSegmentJob`opération. Le code suivant montre comment créer une tâche de segmentation par lots. Donnez un nom à la tâche, spécifiez l'Amazon Resource Name (ARN) de la version de solution à utiliser, spécifiez l'ARN de votre rôle Amazon Personalize IAM et spécifiez le

chemin Amazon S3 vers votre fichier d'entrée et les emplacements de sortie. Votre rôle de service IAM doit disposer d'un accès en lecture et en écriture à vos compartiments Amazon S3 d'entrée et de sortie respectivement.

Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Utilisez la syntaxe suivante pour les emplacements d'entrée et de sortie : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** et**`s3://amzn-s3-demo-bucket/<output folder name>/`**.

Spécifiez `numResults` le nombre d'utilisateurs que vous souhaitez qu'Amazon Personalize prédise pour chaque ligne de données d'entrée. La valeur par défaut est 25. Le maximum est de 5 millions. Fournissez éventuellement un `filterArn` pour filtrer les segments d'utilisateurs. Si votre filtre utilise des paramètres d'espace réservé, assurez-vous que les valeurs des paramètres sont incluses dans le JSON d'entrée. Pour de plus amples informations, veuillez consulter Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées).

SDK for Python (Boto3)

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_segment_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Job name",
    numResults = 25,
    filterArn = "Filter ARN",
    roleArn = "IAM service role ARN",
    jobInput =
       {"s3DataSource": {"path": "s3://amzn-s3-demo-bucket/<folder name>/<input JSON
 file name>.json"}},
    jobOutput =
       {"s3DataDestination": {"path": "s3://amzn-s3-demo-bucket/<output folder
 name>/"}}
)
```

SDK for Java 2.x

```
public static String createBatchSegmentJob(PersonalizeClient personalizeClient,
                                                    String solutionVersionArn,
                                                    String jobName,
```

```
                                                    String filterArn,
                                                    int numResults,
                                                    String
s3InputDataSourcePath,
                                                    String
s3DataDestinationPath,
                                                    String roleArn,
                                                    String explorationWeight,
                                                    String
explorationItemAgeCutOff) {

 long waitInMilliseconds = 60 * 1000;
 String status;
 String batchSegmentJobArn;

 try {
     // Set up data input and output parameters.
     S3DataConfig inputSource = S3DataConfig.builder()
             .path(s3InputDataSourcePath)
             .build();
     S3DataConfig outputDestination = S3DataConfig.builder()
             .path(s3DataDestinationPath)
             .build();

     BatchSegmentJobInput jobInput = BatchSegmentJobInput.builder()
             .s3DataSource(inputSource)
             .build();
     BatchSegmentJobOutput jobOutputLocation = BatchSegmentJobOutput.builder()
             .s3DataDestination(outputDestination)
             .build();


     CreateBatchSegmentJobRequest createBatchSegmentJobRequest =
CreateBatchSegmentJobRequest.builder()
             .solutionVersionArn(solutionVersionArn)
             .filterArn(filterArn)
             .jobInput(jobInput)
             .jobOutput(jobOutputLocation)
             .jobName(jobName)
             .numResults(numResults)
             .roleArn(roleArn)
             .build();
```

```
      batchSegmentJobArn =
 personalizeClient.createBatchSegmentJob(createBatchSegmentJobRequest)
              .batchSegmentJobArn();
      DescribeBatchSegmentJobRequest describeBatchSegmentJobRequest =
 DescribeBatchSegmentJobRequest.builder()
              .batchSegmentJobArn(batchSegmentJobArn)
              .build();

      long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

      // wait until the batch segment job is complete.
      while (Instant.now().getEpochSecond() < maxTime) {

          BatchSegmentJob batchSegmentJob = personalizeClient
                  .describeBatchSegmentJob(describeBatchSegmentJobRequest)
                  .batchSegmentJob();

          status = batchSegmentJob.status();
          System.out.println("batch segment job status: " + status);

          if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
              break;
          }
          try {
              Thread.sleep(waitInMilliseconds);
          } catch (InterruptedException e) {
              System.out.println(e.getMessage());
          }
      }
      return batchSegmentJobArn;

  } catch (PersonalizeException e) {
      System.out.println(e.awsErrorDetails().errorMessage());
  }
  return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchSegmentJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
```

```
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch segment job's parameters.

export const createBatchSegmentJobParam = {
  jobName: "NAME",
  jobInput: {
    s3DataSource: {
      path: "INPUT_PATH",
    },
  },
  jobOutput: {
    s3DataDestination: {
      path: "OUTPUT_PATH",
    },
  },
  roleArn: "ROLE_ARN",
  solutionVersionArn: "SOLUTION_VERSION_ARN",
  numResults: 20,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateBatchSegmentJobCommand(createBatchSegmentJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Le traitement de la tâche par lots peut prendre un certain temps. Vous pouvez vérifier l'état d'une tâche en appelant DescribeBatchSegmentJob et en transmettant un `batchSegmentJobArn` comme paramètre d'entrée. Vous pouvez également répertorier toutes les tâches de segmentation par lots Amazon Personalize de votre AWS environnement en appelantListBatchSegmentJobs.

Exemples de formats de sortie de tâches de segmentation par lots

Une tâche de segmentation par lots importe vos données d'entrée par lots depuis un compartiment Amazon S3, utilise la version de votre solution entraînée à l'aide d'une recette USER_SEGMENTATION pour générer des segments d'utilisateurs et exporte les segments vers un compartiment Amazon S3.

Les sections suivantes répertorient des exemples de sortie JSON pour les tâches de segmentation par lots par recette.

Rubriques

- Affinité entre les objets
- Affinité entre les attributs et les objets

Affinité entre les objets

L'exemple suivant montre le format du fichier JSON de sortie pour la recette Item-Affinity.

```
{"input": {"itemId": "105"}, "output": {"recommendedUsers": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedUsers": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedUsers": ["2", "442", "435"]}}
...
```

Affinité entre les attributs et les objets

L'exemple suivant montre le format du fichier JSON de sortie pour la Item-Attribute-Affinity recette.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"", "output":
 {"recommendedUsers": ["25", "78", "108"]}}
{"itemAttributes": "ITEMS.genres = \"Adventure\"", "output": {"recommendedUsers":
 ["87", "31", "129"]}}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"", "output":
 {"recommendedUsers": ["8", "442", "435"]}}
...
```

Recommandations de filtrage et segments d'utilisateurs

Lorsque vous recevez des recommandations par le biais d'un outil de recommandation de domaine ou d'une campagne personnalisée, vous pouvez filtrer les résultats en fonction de critères personnalisés. Par exemple, il se peut que vous ne souhaitiez pas recommander des produits qu'un utilisateur a déjà achetés ou recommander uniquement des articles destinés à un groupe d'âge en particulier.

De même, avec les recettes USER_SEGMENTATION, vous ne souhaiterez peut-être pas inclure certains types d'utilisateurs dans les segments d'utilisateurs. En filtrant vos résultats, vous pouvez contrôler les éléments qui seront recommandés aux utilisateurs ou les utilisateurs qui seront inclus dans les segments d'utilisateurs.

Vous pouvez créer, modifier, supprimer et appliquer des filtres à l'aide de la console Amazon Personalize, du AWS Command Line Interface (AWS CLI) et AWS des SDK.

- Pour les recommandations en temps réel, vous appliquez un filtre et spécifiez les valeurs des paramètres de filtre lorsque vous appelez les GetPersonalizedRanking opérations GetRecommendations GetActionRecommendations, ou. Vous pouvez également appliquer un filtre lorsque vous recevez des recommandations issues d'une campagne ou d'un recommandeur dans la console.

  Lorsque vous recevez des recommandations d'articles en temps réel avec des articles personnalisés ou connexes, des recettes ou des cas d'utilisation, vous pouvez spécifier une promotion dans votre demande. Une promotion utilise un filtre pour définir des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés. Pour plus d’informations, consultez Promouvoir des articles dans le cadre de recommandations en temps réel.
- Pour les flux de travail par lots, vous devez inclure toutes les valeurs de paramètres de filtre dans votre code JSON d'entrée. Vous spécifiez ensuite le Amazon Resource Name (ARN) du filtre lorsque vous créez une tâche d'inférence par lots ou une tâche de segment par lots. Pour plus d’informations, consultez Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées).

Filtrer les mises à jour pour les nouveaux enregistrements

Pour les données que vous importez à l'aide PutActionInteractions des opérations PutEvents OR, Amazon Personalize met à jour tous les filtres du groupe de données avec les nouvelles données quelques secondes après l'importation. Par exemple, si votre filtre supprime les articles achetés des recommandations et que vous enregistrez un événement d'achat pour un utilisateur associé à l' PutEvents opération, cet article sera supprimé des futures recommandations pour cet utilisateur dans les secondes suivant l'enregistrement de l'événement.

Pour toutes les autres données importées en masse ou individuellement, Amazon Personalize met à jour tous les filtres du groupe de données avec les nouvelles données dans les 20 minutes suivant la dernière importation.

Rubriques

- Expressions de filtrage
- Filtrer les recommandations en temps réel
- Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées)

Expressions de filtrage

Pour configurer les filtres, vous devez utiliser une expression de filtre correctement formatée. Les expressions de filtre sont composées d'identifiants de jeu de données et de champs dans`dataset.fieldformat`, ainsi que les opérateurs logiques, les mots clés et les valeurs. Pour les valeurs, vous pouvez spécifier des valeurs fixes ou ajouter des paramètres d'espace réservé, définir les critères de filtre lorsque vous recevez des recommandations.

Vous pouvez utiliser des expressions de filtre pour filtrer les éléments, les utilisateurs ou les actions des recommandations basées sur les données des ensembles de données suivants :

- Interactions entre éléments : vous pouvez utiliser des expressions de filtre pour inclure ou exclure des éléments ou des utilisateurs en fonction des données d'interactions. Par exemple, vous pouvez exclure les articles sur lesquels un utilisateur a déjà cliqué (pour les recommandations d'articles) ou n'inclure que les utilisateurs qui ont évalué des articles (pour la Item-Affinity recette). Pour tous les types de recettes, vous pouvez filtrer uniquement en fonction du type d'événement. Vous ne pouvez pas filtrer en fonction d'autres métadonnées d'interaction, telles que les métadonnées contextuelles. Vous ne pouvez pas utiliser les filtres d'interactions entre articles avec leItem-Attribute-Affinity recette.

Amazon Personalize prend en compte jusqu'à 100 des interactions les plus récentes par utilisateur et par type d'événement. Il s'agit d'un quota ajustable. Vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

- Interactions entre les actions : utilisez des expressions de filtre pour inclure ou exclure les actions avec lesquelles un utilisateur a interagi en fonction du type d'événement. Par exemple, vous pouvez exclure des actions déjà effectuées par un utilisateur. Vous ne pouvez pas filtrer en fonction d'autres métadonnées d'action et d'interaction.

  Amazon Personalize prend en compte jusqu'à 300 des interactions d'action les plus récentes par utilisateur et par type d'événement. Il s'agit d'un quota ajustable. Vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas.
- Éléments : utilisez des expressions de filtre pour inclure ou exclure des éléments en fonction de conditions spécifiques aux articles. Vous ne pouvez pas utiliser de filtres pour inclure ou exclure des éléments en fonction de métadonnées textuelles non structurées, telles que les descriptions de produits. Si votre cas d'utilisation de domaine ou votre recette personnalisée génère des recommandations d'éléments connexes, tels que la Similar-Items recette ou le cas d'utilisation du domaine More Like X, vous pouvez utiliser des expressions de filtre pour inclure ou exclure des éléments en fonction des propriétés de l'élément que vous spécifiez dans votre demande de recommandation.
- Utilisateurs : pour les recommandations d'éléments et d'actions, si vous disposez d'un ensemble de données Utilisateurs, vous pouvez exclure ou inclure des éléments ou des actions en fonction d'un`CurrentUser`. Pour les recommandations personnalisées, les articles populaires et les recommandations d'action, c'est à cet utilisateur que vous recevez des recommandations. Pour les articles connexes, il s'agit d'un utilisateur facultatif que vous pouvez spécifier dans votre demande de recommandation.

  Pour les segments d'utilisateurs, vous pouvez utiliser des expressions de filtre pour inclure ou exclure des utilisateurs des segments d'utilisateurs en fonction d'attributs, tels que`Users.MEMBERSHIP_STATUS`.
- Actions : utilisez des expressions de filtre pour inclure ou exclure des actions en fonction de conditions d'action spécifiques. Amazon Personalize exclut automatiquement les actions basées sur `Action expiration timestamp` des `Repeat frequency` données. Vous ne pouvez pas créer de filtres personnalisés supplémentaires qui filtrent en fonction de ces données.

Pour obtenir la liste complète des éléments d'expression de filtre, voirFiltrer les éléments d'expression. Pour des exemples d'expressions de filtre, voirExemples d'expressions de filtre.

Rubriques

- Directives et exigences
- Structure et éléments de l'expression du filtre
- Exemples d'expressions de filtre

Directives et exigences

Lorsque vous créez une expression de filtre, tenez compte des directives et exigences suivantes :

- Vous ne pouvez pas utiliser de filtres pour inclure ou exclure des éléments en fonction de métadonnées textuelles non structurées, telles que les descriptions de produits.
- Si vous filtrez en fonction des données relatives aux interactions entre éléments ou actions, vous ne pouvez filtrer qu'en fonction du type d'événement. Vous ne pouvez pas filtrer en fonction d'autres métadonnées d'interaction, telles que les métadonnées contextuelles.
- Amazon Personalize ignore les majuscules uniquement lorsqu'il fait correspondre des types d'événements.
- Vous ne pouvez pas utiliser les ensembles de données Item Interaction et Item dans une seule expression. Pour créer un filtre filtrant par interaction puis par ensembles de données d'éléments (ou le contraire), vous devez enchaîner deux expressions ou plus. Pour plus d’informations, consultez Combinaison de plusieurs expressions.
- Vous ne pouvez pas utiliser les ensembles de données d'interaction entre les éléments et d'action dans une seule expression. Pour créer un filtre filtrant par interaction avec un élément puis par ensembles de données Action (ou le contraire), vous devez enchaîner deux expressions ou plus. Pour plus d’informations, consultez Combinaison de plusieurs expressions.
- Vous ne pouvez pas utiliser les filtres d'interactions entre articles avec leItem-Attribute-Affinity recette.
- Vous ne pouvez pas créer d'expressions de filtre qui filtrent à l'aide de valeurs de type booléen dans votre schéma. Pour filtrer en fonction de valeurs booléennes, utilisez un schéma avec un champ de type String et utilisez les valeurs `"True"` et `"False"` dans vos données. Vous pouvez également utiliser le type int ou long et les valeurs `0` et`1`.

- Le nombre maximum de champs de jeu de données distincts pour un filtre, que ce soit dans une expression ou dans plusieurs expressions enchaînées, est de 10. Le nombre maximum de champs de jeu de données distincts pour tous les filtres d'un groupe de jeux de données est de 20.
- Vous pouvez appliquer un filtre à l' CurrentItem élément uniquement si votre cas d'utilisation de domaine ou votre recette personnalisée génère des recommandations relatives à des éléments connexes, tels que la Similar-Items recette ou le cas d'utilisation du domaine More Like X.
- Vous ne pouvez pas utiliser de paramètres d'espace réservé dans une expression de filtre qui utilise l'opérateur NOT_IN. Utilisez plutôt l'opérateur IN et utilisez l'action opposée. Par exemple, utilisez Inclure au lieu d'Exclure (ou l'inverse).
- Vous ne pouvez pas créer de filtres qui filtrent en fonction `Repeat frequency` des données `Action expiration timestamp` et. Amazon Personalize filtre automatiquement les recommandations d'action en fonction de ces données.

Structure et éléments de l'expression du filtre

Cette section contient des informations sur la structure des expressions de filtre et de leurs éléments.

Rubriques

- Structure d'expression du filtre
- Filtrer les éléments d'expression

Structure d'expression du filtre

La structure générale d'une expression de filtre est la suivante :

```
EXCLUDE/INCLUDE ItemID/ActionID/UserID WHERE dataset type.field IN/NOT IN (value/
parameter)
```

Vous pouvez créer manuellement des expressions de filtre ou obtenir de l'aide concernant la syntaxe et la structure des expressions à l'aide du générateur d'expressions de la console.

Filtrer les éléments d'expression

Utilisez les éléments suivants pour créer des expressions de filtre :

INCLURE ou EXCLURE

`INCLUDE`À utiliser pour limiter les recommandations aux seuls éléments qui répondent aux critères du filtre OU `EXCLUDE` pour supprimer tous les éléments qui répondent aux critères du filtre.

ItemID/ActionID/UserID

Utilisez l'un de ces éléments après l'`EXCLUDE`élément `INCLUDE` ou. L'élément que vous utilisez varie selon que vous filtrez les éléments (pour les recommandations d'éléments), les actions (pour les recommandations d'actions) ou les utilisateurs (pour les segments d'utilisateurs).

WHERE

`WHERE`À utiliser pour vérifier les conditions des éléments, des actions ou des utilisateurs. Vous devez utiliser l'`WHERE`élément après le `ItemID``ActionID`, ou`UserID`.

AND/OR

Pour enchaîner plusieurs conditions au sein d'une même expression de filtre, utilisez `AND` ou`OR`. Les conditions sont enchaînées en utilisant `AND` ou ne `OR` peuvent affecter que les champs du jeu de données utilisé dans la première condition.

Dataset.field

Indiquez le jeu de données et le champ de métadonnées selon lesquels vous souhaitez filtrer les recommandations`dataset`. `field`format. Par exemple, pour filtrer les recommandations d'éléments en fonction du champ des genres de votre jeu de données d'articles, vous devez les utiliser Items.genres dans votre expression de filtre.

État IF

Utilisez une `IF` condition uniquement pour vérifier les conditions de la `CurrentUser` et une seule fois à la fin d'une expression. Cependant, vous pouvez étendre une `IF` condition en utilisant`AND`.

CurrentUser.attribute

Pour filtrer les recommandations d'articles en fonction de l'utilisateur pour lequel vous recevez des recommandations, dans une condition IF uniquement, utilisez `CurrentUser` et fournissez le champ utilisateur. Par exemple : `CurrentUser.AGE`.

CurrentItem.attribute

Pour les articles connexes, les recettes et les cas d'utilisation uniquement, utilisez`CurrentItem`. `attribute`pour filtrer les articles en fonction d'un attribut de l'article que vous spécifiez dans

votre demande de recommandations d'articles connexes. Par exemple, `CurrentItem.GENRE` ou `CurrentItem.PRICE`.

Vous pouvez appliquer un filtre à l' CurrentItem élément uniquement si votre cas d'utilisation de domaine ou votre recette personnalisée génère des recommandations relatives à des éléments connexes, tels que la Similar-Items recette ou le cas d'utilisation du domaine More Like X. La première fois que vous créez un filtre avec un `CurrentItem` élément, la création du filtre peut prendre quelques minutes. Si vous l'utilisez AWS KMS pour le chiffrement, la création du filtre peut prendre jusqu'à 15 minutes.

IN/NOT DANS

Utilisez `IN` ou `NOT IN` comme opérateur de comparaison pour filtrer en fonction de la correspondance (ou de la non-correspondance) d'une ou de plusieurs valeurs de chaîne. Amazon Personalize filtre uniquement sur des chaînes exactes.

Opérateurs de comparaison

Utilisez =, <, <=, >, >= et ! = opérateurs pour tester l'égalité des données numériques, y compris les données transmises dans un paramètre d'espace réservé.

Caractère astérisque (*)

*À utiliser pour inclure ou exclure des interactions de tous types. À utiliser * uniquement pour les expressions de filtre qui utilisent le `EVENT_TYPE` champ d'un `Interactions` ensemble de données.

Séparateur de tuyaux

Utilisez le séparateur de tubes (`|`) pour enchaîner plusieurs expressions. Pour plus d'informations, consultez Combinaison de plusieurs expressions.

Parameters

Pour les expressions qui utilisent des opérateurs de comparaison ou l'`IN`opérateur, utilisez le signe dollar ($) et un nom de paramètre pour ajouter un paramètre d'espace réservé en tant que valeur. Par exemple : `$GENRES`. Dans cet exemple, lorsque vous recevez des recommandations, vous indiquez le ou les genres à filtrer.

> **Note**
>
> Vous définissez un nom de paramètre lorsque vous l'ajoutez à une expression. Le nom du paramètre ne doit pas nécessairement correspondre au nom du champ. Nous vous

recommandons d'utiliser un nom de paramètre similaire au nom du champ et facile à mémoriser. Vous utilisez le nom du paramètre (distinguez les majuscules et minuscules) lorsque vous appliquez le filtre aux demandes de recommandations. Pour un exemple montrant comment appliquer un filtre avec des paramètres d'espace réservé lors de l'utilisation des AWS SDK, voir. Appliquer un filtre (AWS SDK)

Exemples d'expressions de filtre

Utilisez les expressions de filtre décrites dans les sections suivantes pour apprendre à créer vos propres expressions de filtre.

Rubriques

- Exemples d'expressions de filtre de recommandation d'éléments
- Expressions de filtrage des segments d'utilisateurs
- Exemples d'expressions de filtres de recommandation d'action
- Combinaison de plusieurs expressions

Exemples d'expressions de filtre de recommandation d'éléments

Les expressions de filtre suivantes montrent comment filtrer les recommandations d'éléments en fonction des interactions entre les éléments, des métadonnées des éléments et des métadonnées utilisateur. Ils sont organisés par type de données.

Rubriques

- Données d'interaction entre les articles
- Données relatives à l'article
- Données de l’utilisateur

Données d'interaction entre les articles

L'expression suivante exclut les éléments en fonction d'un type d'événement (tel qu'un clic) ou des types d'événements que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$EVENT_TYPE` paramètre.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

L'expression suivante exclut les éléments sur lesquels un utilisateur a cliqué ou diffusé en streaming.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click", "stream")
```

L'expression suivante inclut uniquement les éléments sur lesquels l'utilisateur a cliqué.

```
INCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click")
```

Données relatives à l'article

L'expression suivante exclut les éléments en fonction d'une ou de plusieurs catégories que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$CATEGORY` paramètre.

```
EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)
```

L'expression suivante inclut uniquement les articles moins chers que l'article actuel (l'article que vous spécifiez dans la demande de recommandations d'articles connexes) et créés par le même studio que l'article actuel. Vous pouvez appliquer un filtre à l' CurrentItem élément uniquement si votre cas d'utilisation de domaine ou votre recette personnalisée génère des recommandations d'articles connexes.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.GENRE IN
 CurrentItem.GENRE
```

L'expression suivante exclut les éléments en fonction de plusieurs niveaux de champs catégoriels. Il exclut les éléments dont la valeur CATEGORY_L1 n'a pas de `shoe` valeur CATEGORY_L2 égale à. `boot`

```
EXCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("shoe") AND Items.CATEGORY_L2 NOT IN
 ("boot")
```

L'expression suivante inclut uniquement les articles dont le prix est inférieur ou égal au prix que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$PRICE` paramètre.

```
INCLUDE ItemID WHERE Items.PRICE <= $PRICE
```

L'expression suivante inclut uniquement les éléments créés avant un horodatage (à l'époque Unix) que vous spécifiez lorsque vous recevez des recommandations.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

L'expression suivante inclut uniquement les éléments dotés d'un ou de plusieurs genres que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$GENRE` paramètre.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE)
```

L'expression suivante inclut uniquement les éléments plus chers que l'article actuel et créés plus récemment qu'un horodatage (en temps d'époque Unix) que vous spécifiez. Vous pouvez utiliser ce filtre si vous recevez des recommandations d'articles connexes et que vous souhaitez appliquer des règles commerciales spécifiques en fonction du prix et d'une date de création variable.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.CREATION_TIMESTAMP >
 $DATE
```

Données de l’utilisateur

L'expression suivante exclut les éléments dotés d'un ou de plusieurs genres que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$GENRE` paramètre, mais uniquement si l'âge de l'utilisateur actuel est égal à la valeur que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$AGE` paramètre.

```
EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE) IF CurrentUser.AGE = $AGE
```

L'expression suivante inclut uniquement les éléments correspondant à `watch` CATEGORY_L1 et `luxury` à CATEGORY_L2, si l'âge de l'utilisateur actuel est supérieur. 18

```
INCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("watch") AND Items.CATEGORY_L2 IN ("luxury")
 IF CurrentUser.AGE > 18
```

Expressions de filtrage des segments d'utilisateurs

Les expressions de filtre suivantes montrent comment filtrer les segments d'utilisateurs en fonction des données d'interactions entre les éléments et des métadonnées utilisateur. Ils sont organisés par type de données.

Données de l’utilisateur

L'expression de filtre suivante inclut uniquement les utilisateurs dont le statut d'adhésion est égal à la valeur que vous spécifiez lorsque vous obtenez des segments d'utilisateurs.

```
INCLUDE UserID WHERE Users.MEMBERSHIP_STATUS IN ($MEMBERSHIP)
```

L'expression de filtre suivante exclut les utilisateurs dont la valeur est AGE inférieure à une valeur que vous spécifiez lorsque vous obtenez des segments d'utilisateurs.

```
EXCLUDE UserID WHERE Users.AGE < $AGE
```

Données d'interaction entre les articles

L'expression de filtre suivante inclut uniquement les utilisateurs qui ont cliqué ou évalué des éléments.

```
INCLUDE UserID WHERE Interactions.EVENT_TYPE IN ("click", "rating")
```

L'expression de filtre suivante exclut les utilisateurs des segments d'utilisateurs qui interagissent avec un élément avec un type d'événement que vous spécifiez lorsque vous obtenez des segments d'utilisateurs.

```
EXCLUDE UserID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

Exemples d'expressions de filtres de recommandation d'action

Les exemples d'expressions de filtre suivants montrent comment filtrer les actions en fonction des données d'interactions, des données d'action et des données utilisateur. Ils sont organisés par type de données.

Rubriques

- Données d'interaction entre actions
- Données d'action
- Données de l’utilisateur

Données d'interaction entre actions

L'expression de filtre suivante inclut uniquement les actions des recommandations avec lesquelles l'utilisateur a interagi, lorsque ces interactions comportent un type d'événement que vous spécifiez lorsque vous recevez des recommandations.

```
INCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

L'expression de filtre suivante exclut les actions que l'utilisateur n'a pas entreprises en fonction du type d'événement.

```
EXCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ("NOT_TAKEN")
```

Données d'action

L'expression suivante exclut les actions basées sur une ou plusieurs catégories que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$CATEGORY` paramètre.

```
EXCLUDE ActionID WHERE Actions.CATEGORY IN ($CATEGORY)
```

L'expression suivante inclut uniquement les actions dont la valeur est supérieure à celle que vous spécifiez lorsque vous recevez des recommandations.

```
INCLUDE ActionID WHERE Actions.VALUE > ($VALUE)
```

Données de l’utilisateur

L'expression suivante inclut uniquement les actions pour les membres premium si l'utilisateur actuel possède un abonnement premium.

```
INCLUDE ActionID WHERE Action.MEMBERSHIP_LEVEL IN ("Premium") IF CurrentUser.MEMBERSHIP
 = $PREMIUM
```

L'expression suivante exclut les actions dont la valeur est `VALUE` inférieure à la valeur que vous spécifiez lorsque vous recevez des recommandations si l'utilisateur actuel est un membre premium.

```
EXCLUDE ActionID WHERE Actions.VALUE < ($VALUE) IF CurrentUser.MEMBERSHIP = $PREMIUM
```

Combinaison de plusieurs expressions

Pour combiner plusieurs expressions, vous devez utiliser un séparateur de tubes (`|`). Utilisez une combinaison d'expressions lorsque vous souhaitez utiliser un filtre et un filtre uniques sur les jeux de données d'éléments et d'interactions d'éléments, ou sur les ensembles de données d'interactions d'actions et d'actions. Chaque expression est d'abord évaluée indépendamment et le résultat est soit l'union, soit l'intersection des deux résultats. Les exemples suivants montrent comment créer des expressions pour les objets et les ensembles de données d'interactions entre éléments, mais les mêmes règles s'appliquent lorsque vous travaillez avec des actions et des interactions d'actions.

Exemple d'expressions correspondantes

Si les deux expressions utilisent `EXCLUDE` ou si les deux expressions utilisent`INCLUDE`, le résultat est l'union des deux résultats comme suit (A et B sont des expressions différentes) :

- `Exclude A | Exclude B` est égal à `Exclude result from A or result from B`
- `Include A | Include B` est égal à `Include result from A or result from B`

L'exemple suivant montre comment combiner deux expressions qui utilisent`INCLUDE`. La première expression inclut uniquement les éléments associés à une ou plusieurs catégories que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$CATEGORY` paramètre. La deuxième expression inclut les éléments que l'utilisateur a marqués comme tels`favorite`. Les recommandations incluront uniquement les articles de la catégorie que vous spécifiez ainsi que les articles que l'utilisateur a marqués comme favoris.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY) | INCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("favorite")
```

Exemple d'INCLUDE et d'EXCLUDE

Si une ou plusieurs expressions sont utilisées `INCLUDE` et qu'une autre expression est utilisée`EXCLUDE`, le résultat est la `EXCLUDE` soustraction du résultat de l'`INCLUDE`expression comme suit (A, B, C et D sont des expressions différentes).

- `Include A | Exclude B` est égal à `Include result from A - result from B`
- `Include A | Include B | Exclude C | Exclude D` est égal à `Include (A or B) - (C or D)`

L'ordre des expressions n'a pas d'importance : si l'expression EXCLUDE précède l'expression INCLUDE, le résultat est le même.

L'exemple suivant montre comment combiner une `INCLUDE` expression et une `EXCLUDE` expression. La première expression inclut uniquement les éléments associés à un ou plusieurs genres que vous spécifiez lorsque vous recevez des recommandations à l'aide du `$GENRE` paramètre. La deuxième expression exclut les éléments sur lesquels l'utilisateur a cliqué ou diffusé en streaming. Les recommandations incluront uniquement les éléments du genre que vous spécifiez et qui n'ont pas été cliqués ou diffusés en continu.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE) | EXCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("click", "stream")
```

Filtrer les recommandations en temps réel

Vous pouvez filtrer les recommandations en temps réel à l'aide de la console Amazon Personalize AWS Command Line Interface (AWS CLI) ou AWS des SDK.

Lorsque vous recevez des recommandations personnalisées ou des articles similaires, vous pouvez spécifier une promotion dans votre demande. Une promotion utilise un filtre pour définir des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés. Pour plus d’informations, consultez Promouvoir des articles dans le cadre de recommandations en temps réel.

Rubriques

- Filtrer les recommandations en temps réel (console)
- Filtrer les recommandations en temps réel (AWS CLI)
- Filtrer les recommandations en temps réel (AWS SDK)

Filtrer les recommandations en temps réel (console)

Pour filtrer les recommandations en temps réel à l'aide de la console, créez un filtre, puis appliquez-le à une demande de recommandation.

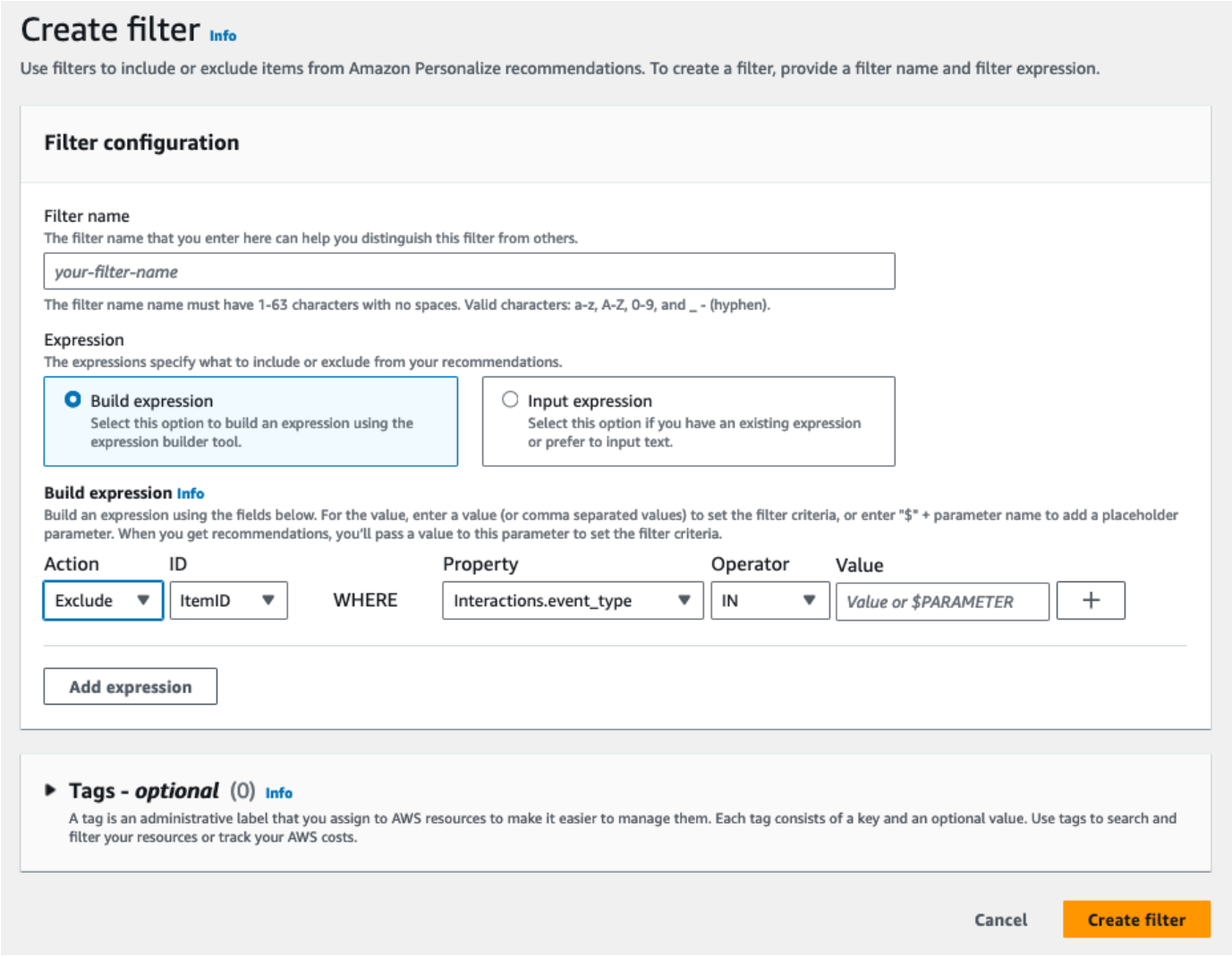
> **Note**
>
> Pour filtrer les recommandations à l'aide d'un filtre paramétré et d'une campagne déployée avant le 10 novembre 2020, vous devez redéployer la campagne en utilisant l'UpdateCampaignopération ou en créer une nouvelle.

Création d'un filtre (console)

Pour créer un filtre dans la console, choisissez le groupe de données contenant la campagne ou le recommandateur que vous souhaitez utiliser pour obtenir des recommandations filtrées. Indiquez ensuite un nom de filtre et une expression de filtre.

Pour créer un filtre (console)

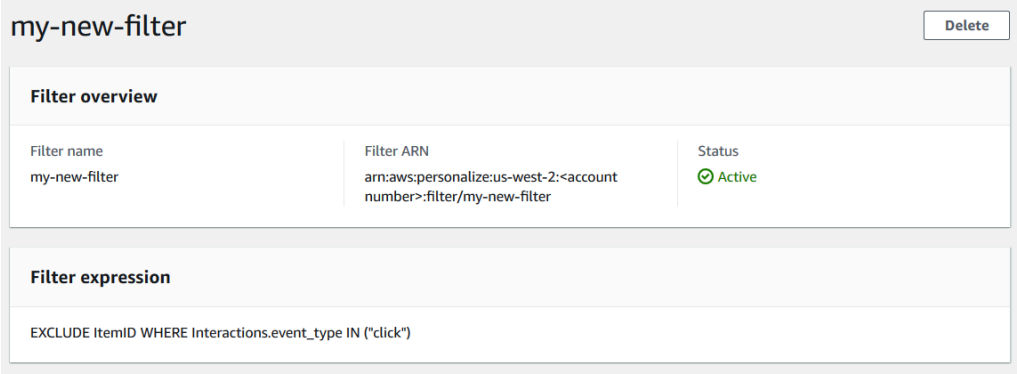
1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne ou le recommandateur que vous souhaitez utiliser pour obtenir des recommandations filtrées.
3. Dans le volet de navigation, choisissez Filtres, puis choisissez Créer un nouveau filtre. La page Créer un filtre s'affiche.



4. Dans Nom du filtre, entrez le nom de votre filtre. Vous choisirez le filtre portant ce nom lorsque vous l'appliquerez à une demande de recommandation.
5. Pour Expression, choisissez Créer une expression ou Ajouter une expression manuellement et créez ou insérez votre expression :

   - Pour utiliser le générateur d'expressions, choisissez Créer une expression. Le générateur d'expressions fournit une structure, des champs et des directives pour créer des expressions de filtre correctement formatées. Pour plus d'informations, consultez Utilisation du générateur d'expressions de filtre.
   - Pour saisir votre propre expression, choisissez Ajouter une expression manuellement. Pour plus d'informations, consultez Filtrer les éléments d'expression.
6. Choisissez Finish (Terminer). La page d'aperçu du filtre indique le nom de ressource Amazon (ARN), le statut et l'expression complète du filtre. Pour supprimer le filtre, choisissez Supprimer.

Pour plus d'informations sur la recherche et la suppression de filtres une fois que vous avez quitté la page d'aperçu, consultezSupprimer un filtre (console).



Appliquer un filtre (console)

Pour appliquer un filtre, dans Test recommender (pour les recommandeurs) ou Test campaign results (pour les campagnes personnalisées), choisissez le filtre et entrez les valeurs des paramètres de filtre. Obtenez ensuite des recommandations pour un utilisateur.

> **Important**
>
> Pour les expressions de filtre qui utilisent un `INCLUDE` élément, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres dont les expressions utilisent un `EXCLUDE` élément, vous pouvez omettre le`filter-values`. Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

Pour appliquer un filtre (console)

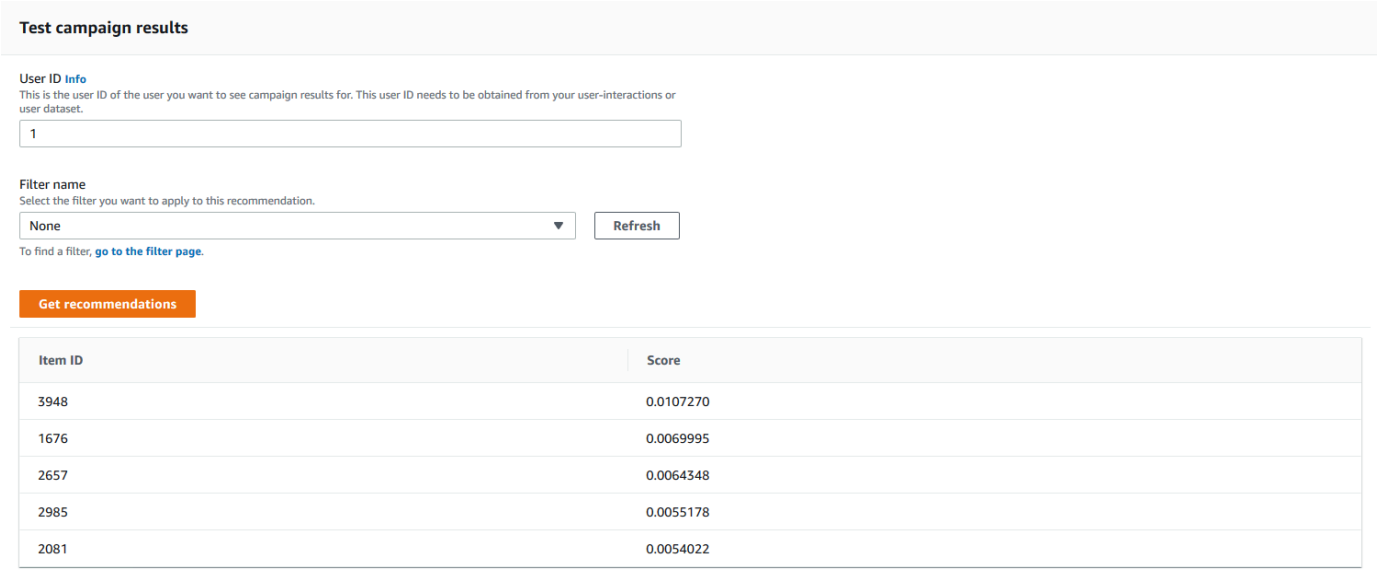
1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez le groupe de jeux de données qui contient la campagne ou le recommandateur que vous souhaitez utiliser pour obtenir des recommandations filtrées.
3. En fonction du type de groupe de données ou de ressource de votre jeu de données, effectuez l'une des opérations suivantes :

   a. Pour un groupe de jeux de données de domaine, dans le volet de navigation, sélectionnez Recommenders.

b. Pour un groupe de jeux de données personnalisé ou des ressources personnalisées, dans le volet de navigation, choisissez Ressources personnalisées puis Campagnes.

4. Sur la page Recommandeurs ou Campagnes, choisissez le recommandeur ou la campagne cible.

5. À titre de comparaison, commencez par obtenir des recommandations sans appliquer de filtre. Sous Test recommander/Test des résultats de la campagne, entrez l'ID d'un utilisateur pour lequel vous souhaitez obtenir des recommandations, ou l'ID de l'article pour les articles connexes, puis choisissez Obtenir des recommandations. Un tableau contenant les principales recommandations s'affiche.



6. Dans le menu Nom du filtre, choisissez le filtre que vous avez créé. Si votre filtre comporte des paramètres d'espace réservé, les champs associés à chaque paramètre apparaissent.

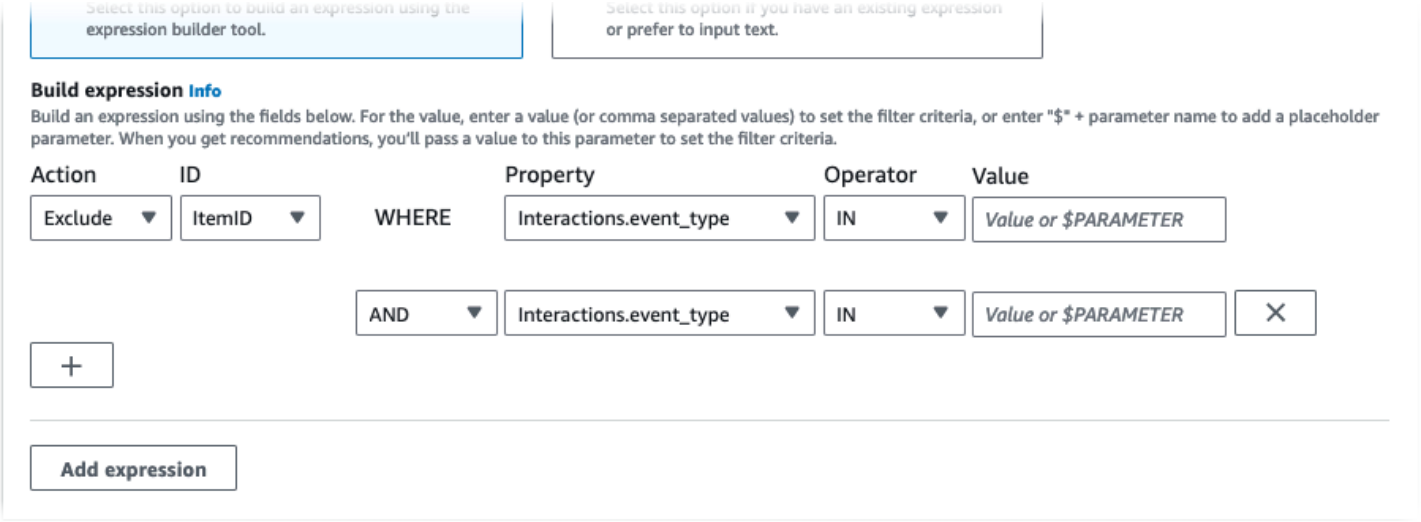
7. Si vous utilisez un filtre avec des paramètres d'espace réservé, entrez la valeur pour chaque paramètre afin de définir les critères du filtre. Pour utiliser plusieurs valeurs pour un paramètre, séparez chaque valeur par une virgule.

8. En utilisant la même étape `User ID` ou `Item ID` comme à l'étape précédente, choisissez Obtenir des recommandations. Le tableau des recommandations s'affiche.

   Par exemple, si l'utilisateur a déjà acheté un article recommandé, le filtre le supprime de la liste des recommandations. Dans cet exemple, les articles 2657 et 2985 ont été remplacés par les articles les plus appropriés que l'utilisateur n'a pas achetés (articles 2641 et 1573).

Utilisation du générateur d'expressions de filtre

Le générateur d'expressions de la page Créer un filtre fournit une structure, des champs et des directives pour créer un filtre correctement formaté.



Pour créer une expression de filtre :

- Utilisez les champs Type, Action, Propriété, Opérateur et Valeur pour créer une expression.

  Pour la valeur, entrez une valeur fixe ou, pour définir des critères de filtre lorsque vous recevez des recommandations, entrez $ + un nom de paramètre. Par exemple : `$GENRES`. Lorsque vous recevez des recommandations, vous fournissez la ou les valeurs à utiliser pour filtrer. Dans cet exemple, vous devez fournir un genre ou une liste de genres lorsque vous recevez des recommandations.

  Séparez les valeurs non paramétriques multiples par une virgule. Vous ne pouvez pas ajouter de paramètres séparés par des virgules à un filtre.

  > **Note**
  >
  > Une fois que vous avez choisi une propriété (au `dataset.field` format), la valeur de propriété pour toutes les lignes suivantes enchaînées par `AND` ou `OR` conditions doit utiliser la même valeur. `dataset`

- Utilisez les boutons + et X pour ajouter ou supprimer une ligne dans votre expression. Vous ne pouvez pas supprimer la première ligne.

- Pour les nouvelles lignes, utilisez les AND OR opérateursIF, ou du menu ET pour créer une chaîne de conditions.

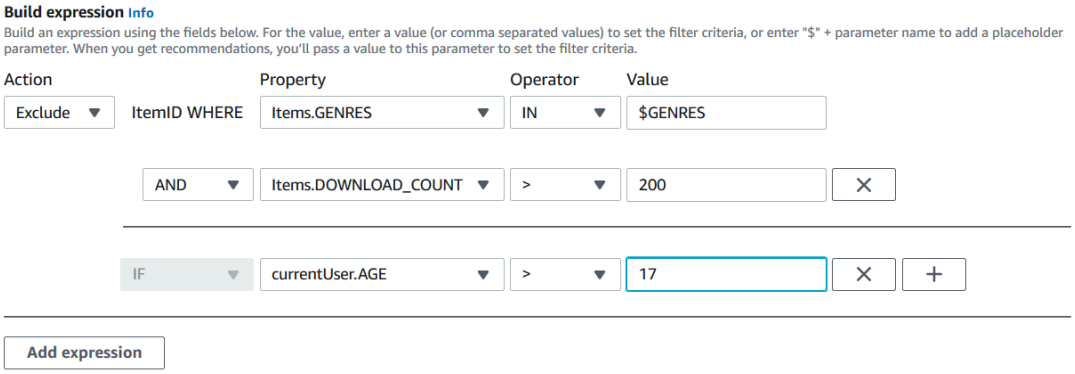
  Pour IF les conditions :

  - Chaque expression ne peut contenir qu'un seul IF élément. Si vous supprimez une condition IF, le générateur d'expressions supprime toutes les AND conditions qui la suivent.
  - Vous ne pouvez utiliser IF des conditions que pour les expressions filtrées selon leCurrentUser.
- Cliquez sur le bouton Ajouter une expression pour ajouter une expression de filtre supplémentaire pour un filtrage plus précis. Chaque expression est d'abord évaluée indépendamment et le résultat est une union des deux résultats.

> **Note**
>
> Pour créer un filtre qui utilise à la fois des ensembles de données d'interactions entre éléments et éléments, ou des jeux de données d'interactions entre actions et actions, vous devez utiliser plusieurs expressions.

Exemple de générateur d'expressions

L'exemple suivant montre comment créer un filtre qui exclut les éléments d'un genre que vous spécifiez lorsque vous recevez des recommandations (notez le paramètre d'espace réservé $GENRES). Le filtre exclut également les éléments dont DOWNLOAD_COUNT la valeur est supérieure à200, mais uniquement si l'âge de l'utilisateur actuel est supérieur à17.



Supprimer un filtre (console)

La suppression d'un filtre supprime le filtre de la liste des filtres d'un groupe de jeux de données.

> **Important**
>
> Vous ne pouvez pas supprimer un filtre pendant qu'une tâche d'inférence par lots est en cours.

Pour supprimer un filtre (console)

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Dans la liste des groupes de jeux de données, choisissez le groupe de jeux de données qui contient le filtre que vous souhaitez supprimer.
3. Dans le volet de navigation, sélectionnez Filtres.
4. Dans la liste des filtres, choisissez le filtre que vous souhaitez supprimer, puis sélectionnez Afficher les détails. La page de détails du filtre apparaît.
5. Choisissez Supprimer et confirmez la suppression dans la boîte de dialogue de confirmation.

Filtrer les recommandations en temps réel (AWS CLI)

Pour filtrer les recommandations à l'aide du AWS CLI, vous créez un filtre, puis vous l'appliquez en spécifiant l'ARN du filtre dans une GetPersonalizedRanking demande GetRecommendations or.

> **Important**
>
> Pour filtrer les recommandations à l'aide d'un filtre paramétré et d'une campagne que vous avez déployée avant le 10 novembre 2020, vous devez redéployer la campagne en utilisant l'UpdateCampaignappel ou en créer une nouvelle.

Création d'un filtre (AWS CLI)

Utilisez l'`create-filter`opération suivante pour créer un filtre et spécifier l'expression du filtre.

Remplacez le `Filter name` par le nom du filtre, puis `Dataset group ARN` par le Amazon Resource Name (ARN) du groupe de jeux de données. Remplacez l'exemple `filter-expression` par votre propre expression de filtre.

```
aws personalize create-filter \
```

```
  --name Filter name \
  --dataset-group-arn dataset group arn \
  --filter-expression "EXCLUDE ItemID WHERE Items.CATEGORY IN (\"$CATEGORY\")"
```

En cas de succès, l'ARN du filtre s'affiche. Enregistrez-le pour une utilisation ultérieure. Pour vérifier que le filtre est actif, utilisez l'DescribeFilteropération avant d'utiliser le filtre.

Pour plus d'informations sur l'API, consultez CreateFilter. Pour plus d'informations sur les expressions de filtre, notamment des exemples, consultezStructure et éléments de l'expression du filtre.

Appliquer un filtre (AWS CLI)

Lorsque vous utilisez les `get-personalized-ranking` opérations `get-action-recommendations` ou`get-recommendations`, vous appliquez un filtre en transmettant les valeurs du filtre `filter-arn` et toutes les valeurs du filtre en tant que paramètres.

Voici un exemple de l'`get-recommendations`opération. Remplacez-le `Campaign ARN` par le Amazon Resource Name (ARN) de votre campagne `User ID` par l'ID de l'utilisateur pour lequel vous recevez des recommandations et `Filter ARN` par l'ARN de votre filtre. Si vous recevez des recommandations d'un recommandeur plutôt que d'une campagne, utilisez à la `recommender-arn` place de `--campaign-arn` et fournissez l'ARN du recommandeur.

Si votre expression comporte des paramètres, incluez l'`filter-values`objet. Pour chaque paramètre de votre expression de filtre, indiquez le nom du paramètre (en distinguant majuscules et minuscules) et les valeurs. Par exemple, si votre expression de filtre comporte un `$GENRE` paramètre, indiquez « GENRE » comme clé et un ou plusieurs genres`"Comedy"`, tels que la valeur. Séparez les valeurs multiples par une virgule. Par exemple : `"\"comedy\",\"drama\",\"horror"\"`.

> **Important**
>
> Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des éléments, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des éléments, vous pouvez omettre le`filter-values`. Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

```
aws personalize-runtime get-recommendations \
```

```
  --campaign-arn Campaign ARN \
  --user-id User ID \
  --filter-arn Filter ARN \
  --filter-values '{
      "Parameter name": "\"value\"",
      "Parameter name": "\"value1\",\"value2\",\"value3\""
    }'
```

Supprimer un filtre (AWS CLI)

Pour supprimer un filtre, procédez comme suit`delete-filter`. Remplacez `filter ARN` par l'ARN du filtre.

```
aws personalize delete-filter --filter-arn Filter ARN
```

Filtrer les recommandations en temps réel (AWS SDK)

Pour filtrer les recommandations à l'aide AWS des SDK, vous créez un filtre, puis vous l'appliquez en spécifiant l'ARN du filtre dans une GetPersonalizedRanking demande GetRecommendations or.

> **Important**
>
> Pour filtrer les recommandations à l'aide d'un filtre paramétré et d'une campagne que vous avez déployée avant le 10 novembre 2020, vous devez redéployer la campagne en utilisant l'UpdateCampaignappel ou en créer une nouvelle.

Création d'un filtre (AWS SDK)

Créez un nouveau filtre avec cette CreateFilter opération. Le code suivant montre comment créer un filtre. Spécifiez le nom du filtre, le nom Amazon Resource Name (ARN) de votre groupe de jeux de données et indiquez votre expression de filtre.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_filter(
    name = 'Filter Name',
```

```
    datasetGroupArn = 'Dataset Group ARN',
    filterExpression = 'EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)'
)
filter_arn = response["filterArn"]
print("Filter ARN: " + filter_arn)
```

SDK for Java 2.x

```
public static String createFilter(PersonalizeClient personalizeClient,
                                  String filterName,
                                  String datasetGroupArn,
                                  String filterExpression) {
    try {
        CreateFilterRequest request = CreateFilterRequest.builder()
                .name(filterName)
                .datasetGroupArn(datasetGroupArn)
                .filterExpression(filterExpression)
                .build();

        return personalizeClient.createFilter(request).filterArn();
    }
    catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateFilterCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the filter's parameters.
export const createFilterParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  name: "NAME" /* required */,
  filterExpression: "FILTER_EXPRESSION" /*required */,
};
```

```
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateFilterCommand(createFilterParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Enregistrez l'ARN du filtre pour une utilisation ultérieure. Pour vérifier que le filtre est actif, utilisez l'DescribeFilteropération avant d'utiliser le filtre. Pour plus d'informations sur l'API, consultez CreateFilter. Pour plus d'informations sur les expressions de filtre, notamment des exemples, consultezStructure et éléments de l'expression du filtre.

Appliquer un filtre (AWS SDK)

Lorsque vous utilisez les GetPersonalizedRanking opérations GetRecommendations GetActionRecommendations, ou, appliquez un filtre en transmettant a `filterArn` et toutes les valeurs de filtre en tant que paramètres.

Le code suivant montre comment filtrer les recommandations d'articles Amazon Personalize pour un utilisateur. Spécifiez l'ID de l'utilisateur pour lequel vous souhaitez obtenir des recommandations, le nom de ressource Amazon (ARN) de votre campagne et l'ARN de votre filtre. Si vous recevez des recommandations d'un recommandeur plutôt que d'une campagne, utilisez à la `recommenderArn` place de `campaignArn` et fournissez l'ARN du recommandeur.

En effet`filterValues`, pour chaque paramètre facultatif de votre expression de filtre, indiquez le nom du paramètre (en distinguant majuscules et minuscules) et la ou les valeurs. Par exemple, si votre expression de filtre comporte un `$GENRES` paramètre, indiquez « GENRES » comme clé, et un ou plusieurs genres, tels que`"\"Comedy"\"`, comme valeur. Pour les valeurs multiples, séparez-les par une virgule. Par exemple : `"\"comedy\",\"drama\",\"horror\""`.

> **Important**
>
> Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des éléments, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour

les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des éléments, vous pouvez omettre le`filter-values`. Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

SDK for Python (Boto3)

```
import boto3

personalize_runtime = boto3.client("personalize-runtime")

response = personalize_runtime.get_recommendations(
    campaignArn = "Campaign ARN",
    userId = "User ID",
    filterArn = "Filter ARN",
    filterValues = {
      "Parameter name": "\"value1\"",
      "Parameter name": "\"value1\",\"value2\",\"value3\""
      ....
    }
)
```

SDK for Java 2.x

L'exemple suivant utilise deux paramètres, l'un avec deux valeurs et l'autre avec une valeur. En fonction de l'expression de votre filtre, modifiez le code pour ajouter ou supprimer les champs ParameterName et ParameterValue.

```
public static void getFilteredRecs(PersonalizeRuntimeClient
 personalizeRuntimeClient,
                                    String campaignArn,
                                    String userId,
                                    String filterArn,
                                    String parameter1Name,
                                    String parameter1Value1,
                                    String parameter1Value2,
                                    String parameter2Name,
                                    String parameter2Value){

    try {

        Map<String, String> filterValues = new HashMap<>();
```

```
        filterValues.put(parameter1Name, String.format("\"%1$s\",\"%2$s\"",
                parameter1Value1, parameter1Value2));
        filterValues.put(parameter2Name, String.format("\"%1$s\"",
                parameter2Value));

        GetRecommendationsRequest recommendationsRequest =
 GetRecommendationsRequest.builder()
                .campaignArn(campaignArn)
                .numResults(20)
                .userId(userId)
                .filterArn(filterArn)
                .filterValues(filterValues)
                .build();

        GetRecommendationsResponse recommendationsResponse =
 personalizeRuntimeClient.getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();

        for (PredictedItem item: items) {
            System.out.println("Item Id is : "+item.itemId());
            System.out.println("Item score is : "+item.score());
        }
    } catch (PersonalizeRuntimeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here:
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
```

```
  filterArn: "FILTER_ARN" /* required to filter recommendations */,
  filterValues: {
    PROPERTY:
      '"VALUE"' /* Only required if your filter has a placeholder parameter */,
  },
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Supprimer un filtre (AWS Python (SDK)

Utilisez la `delete_filter` méthode suivante pour supprimer un filtre. Remplacez `filter ARN` par l'ARN du filtre.

```
import boto3
personalize = boto3.client("personalize")

response = personalize.delete_filter(
  filterArn = "filter ARN"
)
```

Filtrer les recommandations par lots et les segments d'utilisateurs (ressources personnalisées)

Le filtrage des recommandations par lots et des segments d'utilisateurs fonctionne presque de la même manière que le filtrage des recommandations en temps réel. Il suit le même flux de travail décrit dans Obtenir des recommandations d'articles par lots ouObtenir des segments d'utilisateurs par lots. Pour filtrer les recommandations par lots ou les segments d'utilisateurs, procédez comme suit :

1. Créez un filtre comme vous le feriez pour les recommandations en temps réel. Pour plus d'informations, consultez Filtrer les recommandations en temps réel.
2. Préparez vos données d'entrée et chargez-les sur Amazon S3 comme décrit dans Préparation des données d'entrée pour les recommandations par lots ouPréparation des données d'entrée pour les segments d'utilisateurs. Si votre filtre utilise des paramètres d'espace réservé, vous devez ajouter un `filterValues` objet supplémentaire. Pour plus d'informations, consultez Fournir des valeurs de filtre dans votre JSON d'entrée. Si votre filtre n'utilise pas de paramètres d'espace réservé, vos données d'entrée peuvent suivre les exemples fournis dans Exemples JSON d'entrée et de sortie de tâches d'inférence par lots Exemples JSON d'entrée et de sortie de tâches de segmentation par lots
3. Créez un emplacement distinct pour vos données de sortie, qu'il s'agisse d'un dossier ou d'un autre compartiment Amazon S3.
4. Créez une tâche d'inférence par lots ou une tâche de segmentation par lots. Lorsque vous créez la tâche, spécifiez l'Amazon Resource Name (ARN) de votre filtre.
5. Lorsque le travail d'inférence ou de segmentation par lots est terminé, récupérez les recommandations ou les segments utilisateur depuis votre emplacement de sortie dans Amazon S3.

Rubriques

- Fournir des valeurs de filtre dans votre JSON d'entrée
- Filtrer les flux de travail par lots (console)
- Filtrer les flux de travail par lots (AWS SDK)

Fournir des valeurs de filtre dans votre JSON d'entrée

Pour les filtres comportant des paramètres d'espace réservé, par exemple`$GENRE`, vous devez fournir les valeurs des paramètres d'un `filterValues` objet dans votre code JSON d'entrée. Pour un `filterValues` objet, chaque clé est un nom de paramètre. Chaque valeur correspond au critère que vous transmettez en tant que paramètre. Entourez chaque valeur de guillemets évadés :`"filterValues":{"GENRES":"\"drama\""}`. Pour plusieurs valeurs, séparez chaque valeur par une virgule : `"filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}`

Exemple JSON de saisie d'une tâche d'inférence par lots

Voici un exemple des premières lignes d'un fichier d'entrée JSON pour une tâche d'inférence par lots. L'exemple inclut l'`filterValues`objet. La `GENRES` clé correspond à un `$GENRES` espace réservé dans l'expression du filtre. Dans cet exemple, la tâche utilise la User-Personalization recette. Pour les recettes RELATED_ITEMS, fournissez un ItemID au lieu du UserId. Pour les recettes PERSONALIZED_RANKING, fournissez l'ID utilisateur et une ItemList.

```
{"userId": "5","filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}}
{"userId": "3","filterValues":{"GENRES":"\"horror\",\"comedy\""}}
{"userId": "34","filterValues":{"GENRES":"\"drama\""}}
```

Pour plus d'exemples de tâches d'inférence par lots, saisissez des données par recette, voirExemples JSON d'entrée et de sortie de tâches d'inférence par lots. Vous pouvez utiliser ces exemples comme point de départ et ajouter l'`filterValues`objet de l'exemple ci-dessus.

Exemple JSON de saisie de tâche par segment par lots

Voici un exemple des premières lignes d'un fichier d'entrée JSON avec des valeurs de filtre pour une tâche de segmentation par lots. La `GENRES` clé correspond à un `$GENRES` espace réservé dans l'expression du filtre.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action
\"","filterValues":{"COUNTRY":"\"Japan\""}}
{"itemAttributes": "ITEMS.genres = \"Horror\"","filterValues":{"COUNTRY":"\"United
 States\"\""}}
{"itemAttributes": "ITEMS.genres = \"Action\" AND ITEMS.genres = \"Adventure
\"","filterValues":{"COUNTRY":"\"England\""}}
```

Pour plus d'exemples de tâches d'inférence par lots, saisissez des données par recette, voirExemples JSON d'entrée et de sortie de tâches de segmentation par lots. Vous pouvez utiliser ces exemples comme point de départ et ajouter l'`filterValues`objet de l'exemple ci-dessus.

Filtrer les flux de travail par lots (console)

Pour filtrer les flux de travail par lots à l'aide de la console Amazon Personalize, vous créez un filtre, puis vous créez une tâche d'inférence par lots ou une tâche de segmentation par lots et vous choisissez le filtre. Pour obtenir des instructions détaillées étape par étape, reportez-vous Création

d'une tâche d'inférence par lots (console) aux sections etCréation d'une tâche de segmentation par lots (console).

Filtrer les flux de travail par lots (AWS SDK)

Pour filtrer les recommandations par lots à l'aide AWS des SDK, créez un filtre et incluez le `FilterArn` paramètre dans la CreateBatchSegmentJob demande CreateBatchInferenceJob ou.

Le code suivant montre comment créer une tâche d'inférence par lots avec un filtre utilisant le AWS SDK pour Python (Boto3). Nous vous recommandons d'utiliser un autre emplacement pour vos données de sortie (soit un dossier, soit un autre compartiment Amazon S3). Pour une explication complète de tous les champs, voirCréation d'une tâche d'inférence par lots ()AWS SDKs.

```
import boto3

personalize = boto3.client("personalize")

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM role ARN",
    filterArn = "Filter ARN",
    jobInput =
        {"s3DataSource": {"path": "S3 input path"}},
    jobOutput =
        {"S3DataDestination": {"path": "S3 output path"}}
)
```

Enregistrement des événements en temps réel pour influencer les recommandations

Un événement est une interaction entre un utilisateur et votre catalogue. Il peut s'agir d'une interaction avec un article, par exemple lorsqu'un utilisateur achète un article ou regarde une vidéo, ou d'une action, telle qu'une demande de carte de crédit ou l'inscription à un programme d'adhésion.

Amazon Personalize peut émettre des recommandations basées uniquement sur des données d'événements en temps réel, sur des données d'événements historiques uniquement ou sur une combinaison des deux. Enregistrez les événements en temps réel au fur et à mesure que vos clients interagissent avec les recommandations. Cela permet de développer vos données d'interactions et de les actualiser. Il informe également Amazon Personalize des centres d'intérêt actuels de votre utilisateur, ce qui peut améliorer la pertinence des recommandations.

Si le cas d'utilisation de votre domaine ou votre recette personnalisée prend en charge la personnalisation en temps réel, Amazon Personalize utilise des événements en temps réel pour mettre à jour et adapter les recommandations en fonction de l'évolution des intérêts de l'utilisateur.

La façon dont vous enregistrez les événements en temps réel dépend du type de données d'interactions que vous importez :

- Pour les interactions entre objets, vous enregistrez les événements en temps réel grâce à l'opération de l'PutEventsAPI. Amazon Personalize ajoute ces données aux données d'interaction entre les articles de votre groupe de jeux de données. Pour de plus amples informations, veuillez consulter Enregistrement des événements d'interaction avec les objets en temps réel.
- Pour les interactions d'action, vous enregistrez les événements en temps réel avec le fonctionnement de l'PutActionInteractionsAPI. Amazon Personalize ajoute ces données au jeu de données Action interactions de votre groupe de jeux de données. Seules les recettes PERSONALIZED_ACTIONS utilisent les données d'interactions entre actions. Pour de plus amples informations, veuillez consulter Enregistrement des événements d'interaction en temps réel.

Rubriques

- Comment les événements en temps réel influencent les recommandations
- Enregistrement des événements d'interaction avec les objets en temps réel
- Enregistrement des événements d'interaction en temps réel

- Enregistrement d'événements pour les utilisateurs anonymes
- Services tiers de suivi des événements
- Exemples d’implémentations

Comment les événements en temps réel influencent les recommandations

Si votre recette prend en charge la personnalisation en temps réel, une fois que vous avez créé un programme de recommandation ou une campagne personnalisée, Amazon Personalize utilise les nouvelles données d'événement enregistrées pour les articles ou les actions existants quelques secondes après l'importation. Les exemples d'utilisation et les recettes suivants prennent en charge la personnalisation en temps réel :

- Recommandé pour vous (cas d'utilisation du commerce électronique)
- Les meilleurs choix pour vous (cas d'utilisation de VIDEO_ON_DEMAND)
- Recette de personnalisation par l'utilisateur V2
- Recette de personnalisation par l'utilisateur
- Recette de classement V2 personnalisée
- Recette de classement personnalisé
- Next-Best-Action recette

Si vous utilisez la recette Trending-Now, Amazon Personalize prend automatiquement en compte les éléments issus des nouvelles données d'événements sur des intervalles configurables. Il n'est pas nécessaire de créer une nouvelle version de solution. Pour de plus amples informations, veuillez consulter Recette Trending-Now.

Si l'article, l'action ou l'utilisateur de l'événement est nouveau, la manière dont Amazon Personalize utilise les données dépend de votre cas d'utilisation ou de votre recette. Pour de plus amples informations, veuillez consulter Mise à jour des données dans les ensembles de données après l'entraînement.

Enregistrement des événements d'interaction avec les objets en temps réel

Un événement d'interaction avec un article est une interaction entre un utilisateur et un article de votre catalogue. Par exemple, un utilisateur qui achète des chaussures ou regarde un film.

Enregistrez les événements d'interaction avec les articles en temps réel lorsque vous montrez les recommandations d'articles à vos clients. Cela permet de développer vos données d'interactions et de les actualiser. Il informe également Amazon Personalize des centres d'intérêt actuels de votre utilisateur, ce qui peut améliorer la pertinence des recommandations.

Vous enregistrez les événements d'interaction entre les éléments avec l'opération d'PutEventsAPI. Amazon Personalize ajoute les données de l'événement au jeu de données sur les interactions entre articles dans votre groupe de jeux de données. Si vous enregistrez deux événements avec exactement le même horodatage et les mêmes propriétés, Amazon Personalize ne conserve qu'un seul des événements. Vous pouvez enregistrer les événements d'interaction avec des objets à l'aide de AWS SDKs, AWS Amplify ou AWS Command Line Interface ()AWS CLI.

Si vous utilisez Apache Kafka, vous pouvez utiliser le connecteur Kafka pour Amazon Personalize afin de diffuser les interactions entre les articles en temps réel vers Amazon Personalize. Pour plus d'informations, consultez Kafka Connector pour Amazon Personalize dans le référentiel personalize-kafka-connectorGithub.

AWS Amplify inclut une JavaScript bibliothèque pour enregistrer les événements d'interaction entre les éléments à partir d'applications clientes Web et une bibliothèque pour enregistrer les événements dans le code du serveur. Pour plus d'informations, consultez la documentation Amplify.

Rubriques

- Exigences relatives à l'enregistrement des événements d'interaction avec les objets et à la formation d'un modèle
- Création d'un outil de suivi des événements liés aux interactions entre objets
- Enregistrement d'un événement d'interaction avec un seul élément
- Enregistrement d'événements d'interaction avec plusieurs éléments à l'aide des données de valeur des événements
- Enregistrement des événements d'interaction avec les objets à l'aide des données d'impressions
- Statistiques relatives aux événements et rapports d'attribution

Exigences relatives à l'enregistrement des événements d'interaction avec les objets et à la formation d'un modèle

Pour enregistrer les événements d'interaction avec un objet, vous avez besoin des éléments suivants :

- Un groupe d’ensembles de données incluant un ensemble de données `Item interactions`, qui peut être vide. Si vous avez lu le Tutoriels de démarrage guide, vous pouvez utiliser le même groupe de jeux de données et le même jeu de données que ceux que vous avez créés. Pour plus d'informations sur la création d'un groupe de jeux de données et d'un ensemble de données, consultezImportation de données de formation dans les ensembles de données Amazon Personalize.
- Un dispositif de suivi d’événement.
- Un appel à l'opération PutEvents d'API.
- Si vous utilisez une AWS Lambda fonction pour appeler l' PutEvents opération, le rôle d'exécution de votre fonction doit être autorisé à effectuer l'`personalize:PutEvents`action avec le caractère générique `*` dans l'`Resource`élément.

Vous pouvez commencer avec un jeu de données d'interactions entre éléments vide et, lorsque vous avez enregistré suffisamment de données, entraîner le modèle en utilisant uniquement les nouveaux événements enregistrés. Pour tous les cas d'utilisation (groupes de jeux de données de domaine) et toutes les recettes (groupes de jeux de données personnalisés), vos données d'interactions doivent présenter les éléments suivants avant l'entraînement :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

Création d'un outil de suivi des événements liés aux interactions entre objets

Avant de pouvoir enregistrer des événements d'interaction avec un objet, vous devez créer un outil de suivi des événements d'interaction avec un élément. Un outil de suivi des événements dirige les nouvelles données d'événements vers le jeu de données d'interactions entre éléments de votre groupe de jeux de données.

Vous créez un outil de suivi des événements à l'aide de la console Amazon Personalize ou de l'opération CreateEventTracker API. Vous transmettez en paramètre le Amazon Resource Name (ARN) du groupe de données qui contient le jeu de données d'interactions avec les articles cible. Pour obtenir des instructions sur la création d'un outil de suivi des événements à l'aide de la console Amazon Personalize, consultezCréation d'un outil de suivi des événements (console).

Un outil de suivi d'événements inclut un identifiant de suivi, que vous transmettez en paramètre lorsque vous utilisez l'PutEventsopération. Amazon Personalize ajoute ensuite les nouvelles données d'événement à l'ensemble de données d'interactions entre articles du groupe de données que vous spécifiez dans votre outil de suivi des événements.

> **Note**
>
> Vous ne pouvez créer qu'un seul outil de suivi des événements d'interaction entre éléments pour un groupe de jeux de données.

Python

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_event_tracker(
    name='MovieClickTracker',
    datasetGroupArn='arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup'
)
print(response['eventTrackerArn'])
print(response['trackingId'])
```

L'ARN du suivi des événements et l'identifiant de suivi s'affichent, par exemple :

```
{
    "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
    "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

AWS CLI

```
aws personalize create-event-tracker \
    --name MovieClickTracker \
    --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup
```

L'ARN du suivi des événements et l'identifiant de suivi s'affichent, par exemple :

```
{
    "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
    "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateEventTrackerCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the event tracker's parameters.
export const createEventTrackerParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  name: "NAME" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateEventTrackerCommand(createEventTrackerParam),
    );
    console.log("Success", response);
```

```
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

SDK for Java 2.x

```
public static String createEventTracker(PersonalizeClient personalizeClient,
                                         String eventTrackerName,
                                         String datasetGroupArn) {

    String eventTrackerId = null;
    String eventTrackerArn = null;
    long maxTime = 3 * 60 * 60;
    long waitInMilliseconds = 30 * 1000;
    String status;

    try {
        CreateEventTrackerRequest createEventTrackerRequest =
 CreateEventTrackerRequest.builder()
            .name(eventTrackerName)
            .datasetGroupArn(datasetGroupArn)
            .build();

        CreateEventTrackerResponse createEventTrackerResponse =
            personalizeClient.createEventTracker(createEventTrackerRequest);

        eventTrackerArn = createEventTrackerResponse.eventTrackerArn();
        eventTrackerId = createEventTrackerResponse.trackingId();

        System.out.println("Event tracker ARN: " + eventTrackerArn);
        System.out.println("Event tracker ID: " + eventTrackerId);

        maxTime = Instant.now().getEpochSecond() + maxTime;

        DescribeEventTrackerRequest describeRequest =
 DescribeEventTrackerRequest.builder()
            .eventTrackerArn(eventTrackerArn)
            .build();

        while (Instant.now().getEpochSecond() < maxTime) {
```

```
            status =
 personalizeClient.describeEventTracker(describeRequest).eventTracker().status();
            System.out.println("EventTracker status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return eventTrackerId;
    }
    catch (PersonalizeException e){
        System.out.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return eventTrackerId;
}
```

Enregistrement d'un événement d'interaction avec un seul élément

Après avoir créé un ensemble de données d'interactions entre articles et un outil de suivi d'événements pour votre groupe de jeux de données, vous êtes prêt à enregistrer les événements d'interaction entre articles. L'exemple suivant montre une `PutEvents` opération qui transmet un événement d'interaction avec un élément. Le schéma correspondant est affiché, ainsi qu'une ligne d'exemple tirée du jeu de données d'interactions entre articles.

Votre application génère un code unique `sessionId` lorsqu'un utilisateur visite votre site Web ou utilise votre application pour la première fois. Vous devez utiliser le même `sessionId` dans tous les événements de la session. Amazon Personalize utilise le `sessionId` pour associer des événements à l'utilisateur avant qu'il ne se connecte (c'est anonyme). Pour de plus amples informations, veuillez consulter Enregistrement d'événements pour les utilisateurs anonymes.

La liste des événements est un tableau d'objets Event. Un `eventType` est requis pour chaque événement. Si vous ne disposez pas de données sur le type d'événement, vous pouvez fournir une valeur d'espace réservé pour répondre à cette exigence.

Il `trackingId` provient du système de suivi des événements que vous avez créé dansCréation d'un outil de suivi des événements liés aux interactions entre objets. Les paramètres `userId`, `itemId` et `sentAt` sont mappés aux champs USER_ID, ITEM_ID et TIMESTAMP d'un ensemble de données `Interactions` historiques correspondants. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Colonnes de jeux de données correspondantes

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE
Example data: user123, item-xyz, 1543631760, click
```

Exemple de code

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'USER_ID',
    sessionId = 'session_id',
    eventList = [{
        'sentAt': 1719511760,
        'eventType': 'click',
        'itemId': 'ITEM_ID'
        }]
)
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { PutEventsCommand } from "@aws-sdk/client-personalize-events";
import { personalizeEventsClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeEventsClient = new PersonalizeEventsClient({ region: "REGION"});

// Convert your UNIX timestamp to a Date.
const sentAtDate = new Date(1613443801 * 1000); // 1613443801 is a testing value.
 Replace it with your sentAt timestamp in UNIX format.
```

```
// Set put events parameters.
const putEventsParam = {
  eventList: [
    /* required */
    {
      eventType: "EVENT_TYPE" /* required */,
      sentAt: sentAtDate /* required, must be a Date with js */,
      eventId: "EVENT_ID" /* optional */,
      itemId: "ITEM_ID" /* optional */,
    },
  ],
  sessionId: "SESSION_ID" /* required */,
  trackingId: "TRACKING_ID" /* required */,
  userId: "USER_ID" /* required */,
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutEventsCommand(putEventsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

AWS CLI

```
aws personalize-events put-events \
    --tracking-id tracking_id \
    --user-id USER_ID \
    --session-id session_id \
    --event-list '[{
        "sentAt": 1719511760,
        "eventType": "click",
        "itemId": "ITEM_ID"
      }]'
```

SDK for Java 2.x

```
public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
```

```
                            String trackingId,
                            String sessionId,
                            String userId,
                            String itemId,
                            String eventType) {

    try {
        Event event = Event.builder()
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
 1000))
            .itemId(itemId)
            .eventType(eventType)
            .build();

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(event)
            .build();

        int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
            .sdkHttpResponse()
            .statusCode();
        System.out.println("Response code: " + responseCode);

        } catch (PersonalizeEventsException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
}
```

Enregistrement d'événements d'interaction avec plusieurs éléments à l'aide des données de valeur des événements

Après avoir créé un ensemble de données d'interactions entre articles et un outil de suivi d'événements pour votre groupe de jeux de données, vous êtes prêt à enregistrer les événements d'interaction entre articles. L'exemple suivant montre comment enregistrer des événements d'interaction entre plusieurs éléments avec différents types d'événements et différentes valeurs d'événements.

Lorsque vous configurez une solution, si votre jeu de données d'interactions avec les articles inclut les champs EVENT_TYPE et EVENT_VALUE, vous pouvez définir une valeur spécifique comme seuil pour exclure les enregistrements de l'entraînement. Pour de plus amples informations, veuillez consulter Choix des données d'interaction entre les objets utilisées pour l'entraînement.

Python

```
import boto3
import json

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'user555',
    sessionId = 'session1',
    eventList = [{
        'eventId': 'event1',
        'sentAt': 1553631760,
        'eventType': 'like',
        'properties': json.dumps({
            'itemId': 'choc-panama',
            'eventValue': 4
            })
        }, {
        'eventId': 'event2',
        'sentAt': 1553631782,
        'eventType': 'rating',
        'properties': json.dumps({
            'itemId': 'movie_ten',
            'eventValue': 3
            })
        }]
)
```

AWS CLI

```
aws personalize-events put-events \
    --tracking-id tracking_id \
    --user-id user555 \
    --session-id session1 \
    --event-list '[{
        "eventId": "event1",
```

```
        "sentAt": 1553631760,
        "eventType": "like",
        "properties": "{\"itemId\": \"choc-panama\", \"eventValue\": \"true\"}"
      }, {
        "eventId": "event2",
        "sentAt": 1553631782,
        "eventType": "rating",
        "properties": "{\"itemId\": \"movie_ten\", \"eventValue\": \"4\",
 \"numRatings\": \"13\"}"
      }]'
```

SDK for Java 2.x

```
public static void putMultipleEvents(PersonalizeEventsClient
 personalizeEventsClient,
                             String trackingId,
                             String sessionId,
                             String userId,
                             String event1Type,
                             Float event1Value,
                             String event1ItemId,
                             int event1NumRatings,
                             String event2Type,
                             Float event2Value,
                             String event2ItemId,
                             int event2NumRatings) {

    ArrayList<Event> eventList = new ArrayList<Event>();

    try {
        Event event1 = Event.builder()
            .eventType(event1Type)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
 1000))
            .itemId(event1ItemId)
            .eventValue(event1Value)
            .properties("{\"numRatings\": "+ event1NumRatings +"}")
            .build();

        eventList.add(event1);

        Event event2 = Event.builder()
            .eventType(event2Type)
```

```
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
 1000))
            .itemId(event2ItemId)
            .eventValue(event2Value)
            .properties("{\"numRatings\": "+ event2NumRatings +"}")
            .build();

        eventList.add(event2);

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(eventList)
            .build();

        int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
            .sdkHttpResponse()
            .statusCode();

        System.out.println("Response code: " + responseCode);

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

> **Note**
>
> Les clés de propriété utilisent des noms en notation camel qui correspondent aux champs dans le schéma d'interactions. Par exemple, si le champ « NUM_RATINGS » est défini dans le schéma Interactions, la clé de propriété doit être. `numRatings`

Enregistrement des événements d'interaction avec les objets à l'aide des données d'impressions

Si vous utilisez la Personnalisation par l'utilisateur recette ou si vous ajoutez le champ IMPRESSIONS à votre schéma pour un ensemble de données d'un groupe de données de domaine, vous pouvez enregistrer les données d'impressions dans votre PutEvents opération.

Les impressions sont des listes d'éléments visibles par un utilisateur lorsqu'il interagissait avec un élément en particulier (par exemple, en cliquant ou en regardant). Amazon Personalize utilise les données d'impressions pour guider l'exploration, et les recommandations incluent des éléments présentant moins de données d'interactions ou de pertinence. Pour plus d'informations sur les impressions implicites et explicites qu'Amazon Personalize peut modéliser, consultezDonnées sur les impressions.

> **Important**
>
> Si vous fournissez des données d'impression implicites et explicites contradictoires dans vos `PutEvents` demandes, Amazon Personalize utilise les impressions explicites par défaut.

Pour enregistrer les recommandations Amazon Personalize, vous montrez à votre utilisateur sous forme de données d'impressions, `recommendationId` incluez-les dans votre PutEvents demande et Amazon Personalize calcule les impressions implicites en fonction de vos données de recommandation.

Pour enregistrer manuellement les données d'impressions pour un événement, listez les impressions dans le paramètre `impression` d'entrée de la PutEvents commande. L'exemple de code suivant montre comment inclure a `recommendationId` et an `impression` dans une PutEvents opération avec le SDK pour Python (Boto3) ou le SDK pour Java 2.x. Si vous incluez les deux, Amazon Personalize utilise les impressions explicites par défaut.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'userId',
    sessionId = 'sessionId',
    eventList = [{
        'eventId': 'event1',
        'eventType': 'rating',
        'sentAt': 1553631760,
        'itemId': 'item id',
        'recommendationId': 'recommendation id',
```

```
        'impression': ['itemId1', 'itemId2', 'itemId3']
        }]
)
```

SDK for Java 2.x

Utilisez la `putEvents` méthode suivante pour enregistrer un événement avec des données d'impressions et un RecommendationID. Pour le paramètre impressions, transmettez la liste des ItemIds sous forme de ArrayList.

```
public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
                             String trackingId,
                             String sessionId,
                             String userId,
                             String eventType,
                             Float eventValue,
                             String itemId,
                             ArrayList<String> impressions,
                             String recommendationId) {

    try {
        Event event = Event.builder()
            .eventType(eventType)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
 1000))
            .itemId(itemId)
            .eventValue(eventValue)
            .impression(impressions)
            .recommendationId(recommendationId)
            .build();

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(event)
            .build();

        int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
            .sdkHttpResponse()
            .statusCode();
        System.out.println("Response code: " + responseCode);
```

```
    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Statistiques relatives aux événements et rapports d'attribution

Pour surveiller le type et le nombre d'événements envoyés à Amazon Personalize, utilisez CloudWatch les métriques Amazon. Pour de plus amples informations, veuillez consulter Surveillance d'Amazon Personalize avec Amazon CloudWatch.

Pour générer des CloudWatch rapports qui montrent l'impact des recommandations, créez une attribution métrique et enregistrez les interactions des utilisateurs avec des recommandations en temps réel. Pour plus d'informations sur la création d'une attribution métrique, consultezMesurer l'impact des recommandations d'Amazon Personalize.

Pour chaque événement, incluez l'ID de recommandation des recommandations que vous avez présentées à l'utilisateur. Ou incluez la source de l'événement, telle qu'un tiers. Importez ces données pour comparer différentes campagnes, recommandations et tiers. Vous pouvez importer au maximum 100 sources d'attribution d'événements.

- Si vous fournissez un`recommendationId`, Amazon Personalize détermine automatiquement la campagne ou le recommandateur source et l'identifie dans les rapports d'une colonne EVENT_ATTRIBUTION_SOURCE.
- Si vous fournissez les deux attributs, Amazon Personalize utilise uniquement le`eventAttributionSource`.
- Si vous ne fournissez pas de source, Amazon Personalize l'étiquette `SOURCE_NAME_UNDEFINED` dans les rapports.

Le code suivant montre comment fournir un `eventAttributionSource` pour un événement dans le cadre d'une PutEvents opération.

```
response = personalize_events.put_events(
    trackingId = 'eventTrackerId',
    userId= 'userId',
    sessionId = 'sessionId123',
    eventList = [{
        'eventId': 'event1',
```

```
        'eventType': 'watch',
        'sentAt': '1667260945',
        'itemId': '123',
        'metricAttribution': {
            'eventAttributionSource': 'thirdPartyServiceXYZ'
        }
    }]
)
statusCode = response['ResponseMetadata']['HTTPStatusCode']
print(statusCode)
```

Le code suivant montre comment fournir un événement `recommendationId` pour une PutEvents opération.

```
response = personalize_events.put_events(
    trackingId = 'eventTrackerId',
    userId= 'userId',
    sessionId = 'sessionId123',
    eventList = [{
        'eventId': 'event1',
        'eventType': 'watch',
        'sentAt': '1667260945',
        'itemId': '123',
        'recommendationId': 'RID-12345678-1234-1234-1234-abcdefghijkl'
    }]
)
statusCode = response['ResponseMetadata']['HTTPStatusCode']
print(statusCode)
```

Enregistrement des événements d'interaction en temps réel

Un événement d'interaction d'action est une interaction entre un utilisateur et une action. Par exemple, un utilisateur s'inscrivant à un programme d'adhésion ou demandant une carte de crédit.

Si vous utilisez une recette personnalisée PERSONALIZED_ACTIONS, enregistrez les événements d'interaction en temps réel au fur et à mesure que vos clients interagissent avec les recommandations d'action. Cela permet de développer vos données d'interactions et de les actualiser. Il informe également Amazon Personalize des centres d'intérêt actuels de votre utilisateur, ce qui peut améliorer la pertinence des recommandations. Seules les recettes personnalisées PERSONALIZED_ACTIONS utilisent les données des interactions d'action.

Vous enregistrez les événements d'interaction avec l'action de PutActionInteractions l'API. Amazon Personalize ajoute ces données au jeu de données Action interactions de votre groupe de jeux de données.

Un événement d'interaction avec une action doit avoir un attribut de type d'événement, qui peut être l'un des suivants :

- Taken : enregistre les événements Taken lorsqu'un utilisateur exécute une action recommandée.
- Non prise — Enregistrez les événements non effectués lorsque votre utilisateur fait le choix délibéré de ne pas effectuer l'action après l'avoir consultée. Par exemple, s'ils choisissent Non lorsque vous leur montrez l'action. Les événements Not Taken peuvent indiquer que le client n'est pas intéressé par l'action.
- Consulté — Enregistrez les événements consultés lorsque vous montrez une action à un utilisateur avant qu'il ne fasse le choix d'effectuer ou de ne pas effectuer une action. Amazon Personalize utilise les événements View pour en savoir plus sur les centres d'intérêt de vos utilisateurs. Par exemple, si un utilisateur voit une action mais ne l'exécute pas, il se peut que cette action ne l'intéresse pas à l'avenir.

Vous pouvez enregistrer des événements en temps réel à l'aide de AWS SDKs, ou AWS Command Line Interface (AWS CLI). Si vous enregistrez deux événements avec exactement le même horodatage et les mêmes propriétés, Amazon Personalize ne conserve qu'un seul des événements.

Rubriques

- Exigences relatives à l'enregistrement des événements d'interaction entre actions
- Trouver l'identifiant de votre outil de suivi des événements d'interaction et d'action
- Enregistrement d'un événement d'interaction avec une seule action
- Enregistrement de plusieurs événements d'interaction entre actions

Exigences relatives à l'enregistrement des événements d'interaction entre actions

Pour enregistrer des événements d'interaction en temps réel, vous avez besoin des éléments suivants :

- Un groupe de jeux de données qui inclut un`Action interactions dataset`, qui peut être vide. Pour plus d'informations sur la création d'un groupe de jeux de données et d'un ensemble de

données, consultezImportation de données de formation dans les ensembles de données Amazon Personalize.

- L'identifiant de votre outil de suivi des événements. Vous spécifiez cet ID lors de l' PutActionInteractions opération. Lorsque vous créez un ensemble de données d'interactions d'action, Amazon Personalize crée automatiquement un outil de suivi des événements d'interaction d'action pour vous. Pour de plus amples informations, veuillez consulter Trouver l'identifiant de votre outil de suivi des événements d'interaction et d'action.
- Un appel de l'opération PutActionInteractions.

Trouver l'identifiant de votre outil de suivi des événements d'interaction et d'action

Lorsque vous créez un ensemble de données d'interactions d'action, Amazon Personalize crée automatiquement un outil de suivi des événements d'interaction d'action pour vous. Vous spécifiez l'ID du tracker dans l'opération PutActionInteractions d'API. Amazon Personalize l'utilise pour diriger les nouvelles données vers le jeu de données Action interactions de votre groupe de jeux de données.

Vous trouverez l'identifiant de votre outil de suivi d'événements sur la page détaillée de votre ensemble de données d'interactions Action dans la console Amazon Personalize. Et vous pouvez trouver l'identifiant en appelant l'opération DescribeDataset API. Le code Python suivant imprime l'ID de suivi d'un ensemble de données d'interactions Action.

```
import boto3

personalize = boto3.client(service_name='personalize')

response = personalize.describe_dataset(
  datasetArn="Action interactions dataset ARN"
)

print(response['trackingId'])
```

Enregistrement d'un événement d'interaction avec une seule action

Après avoir créé un jeu de données d'interactions d'action, vous êtes prêt à enregistrer les événements d'interaction d'action avec l'PutActionInteractionsopération. Le code suivant montre une

`PutActionInteractions` opération qui transmet un événement TAKEN. Vous pouvez enregistrer cet événement lorsque vous montrez à un utilisateur les recommandations d'Amazon Personalize et qu'il entreprend une action, telle qu'une demande de carte de crédit.

`actionInteractions`Il s'agit d'un ensemble d' ActionInteraction objets. Il `trackingId` provient du système de suivi d'événements Amazon Personalize créé lorsque vous avez créé votre ensemble de données d'interactions Action. Pour de plus amples informations, veuillez consulter Trouver l'identifiant de votre outil de suivi des événements d'interaction et d'action.

Votre application génère un code unique `sessionId` lorsqu'un utilisateur visite votre site Web ou utilise votre application pour la première fois. Vous devez utiliser le même `sessionId` dans tous les événements de la session. Amazon Personalize utilise le `sessionId` pour associer des événements à l'utilisateur avant qu'il ne se connecte (c'est anonyme). Pour de plus amples informations, veuillez consulter Enregistrement d'événements pour les utilisateurs anonymes.

Les `sentAt` paramètres `userIdactionId`, et correspondent aux champs USER_ID, ACTION_ID, EVENT_TYPE et TIMESTAMP du jeu de données Action interactions.

Ensemble de données d'interactions Action correspondant

```
USER_ID, ACTION_ID, TIMESTAMP, EVENT_TYPE
user123, action-xyz, 1543631760, TAKEN
```

Exemple de code

AWS CLI

```
aws personalize-events put-action-interactions \
--tracking-id 12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx \
--action-interactions '[{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543631760,
  "eventType": "TAKEN",
  "actionId": "action-xyz"}]'
```

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')
```

```
response = personalize_events.put_action_interactions(
  trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
  actionInteractions=[{
    'userId': 'user123',
    'sessionId': 'abcdefg',
    'timestamp': 1543631760,
    'eventType': 'Taken',
    'actionId': 'action-xyz'
  }]
)
```

Enregistrement de plusieurs événements d'interaction entre actions

Le code suivant montre comment enregistrer plusieurs événements d'interaction d'action pour le même utilisateur avec le même SessionId.

Ensemble de données d'interactions Action correspondant

```
USER_ID, ACTION_ID, EVENT_TYPE, TIMESTAMP
user123, action123, Taken, 1543531139
user123, action345, Not Taken, 1543531139
```

AWS CLI

```
aws personalize-events put-action-interactions \
--tracking-id 6ddfe6b7-cd83-4dd4-b09d-4c35ecbacfe1 \
--action-interactions '[{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Taken",
  "actionId": "action123"
},
{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Not Taken",
  "actionId": "action345"}]'
```

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

response = personalize_events.put_action_interactions(
  trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
  actionInteractions=[{
    'userId': 'user123',
    'sessionId': 'abcdefg',
    'timestamp': 1697848587,
    'eventType': 'Taken',
    'actionId': 'action123'
  },
  {
    'userId': 'user123',
    'sessionId': 'abcdefg',
    'timestamp': 1697848622,
    'eventType': 'Not Taken',
    'actionId': 'action345'
  }]
)
```

Enregistrement d'événements pour les utilisateurs anonymes

> **Important**
>
> Si vous n'enregistrez pas au moins un événement avec `sessionId` et `userId` pour un utilisateur, Amazon Personalize n'utilisera pas l'activité suivie uniquement `sessionId` lors de l'entraînement. Et une fois la formation terminée, les recommandations ne seront plus basées sur l'activité suivie par le`sessionId`. Cela créera un historique continu des événements pour les UserIds avant et après leur connexion.

Vous pouvez enregistrer les événements d'interaction entre des éléments ou des actions pour les utilisateurs avant qu'ils ne créent un compte. Enregistrez les événements pour les utilisateurs anonymes afin de créer un historique continu des événements avec les événements survenus avant

et après leur connexion. Cela fournit à Amazon Personalize davantage de données d'interactions sur l'utilisateur, ce qui peut aider à générer des recommandations plus pertinentes.

Pour enregistrer des événements pour des utilisateurs anonymes (utilisateurs qui ne se sont pas connectés), pour chaque événement, spécifiez uniquement un`sessionId`. Votre application génère un code unique `sessionId` lorsqu'un utilisateur visite votre site Web ou utilise votre application pour la première fois. Vous devez utiliser le même `sessionId` dans tous les événements de la session. Amazon Personalize utilise le `sessionId` pour associer des événements à l'utilisateur avant qu'il ne se connecte.

Amazon Personalize n'utilise pas les événements d'utilisateurs anonymes lors de la formation tant que vous ne les associez pas à un`userId`. Pour de plus amples informations, veuillez consulter Création d'un historique continu des événements pour les utilisateurs anonymes.

Pour fournir une personnalisation en temps réel aux utilisateurs anonymes, spécifiez le SessionId comme UserID dans votre demande ou. GetRecommendations GetActionRecommendations

- Pour des exemples de code montrant comment enregistrer les événements d'interaction entre les éléments à l'aide de l' PutEvents opération et d'un SessionId et d'un UserID, voir. Enregistrement d'un événement d'interaction avec un seul élément
- Pour des exemples de code montrant comment enregistrer des événements d'interaction avec l' PutActionInteractions opération et un SessionId et un UserID, voir. Enregistrement d'un événement d'interaction avec une seule action

Création d'un historique continu des événements pour les utilisateurs anonymes

Pour créer un historique des événements pour un utilisateur anonyme et demander à Amazon Personalize d'utiliser ses événements lors de la formation, enregistrez au moins un événement avec un `sessionId` et un`userId`. Vous pouvez ensuite enregistrer autant d'événements que vous le souhaitez pour le`userId`. Une fois que vous avez commencé à fournir un`userId`, `sessionId` ils peuvent changer. Lors de la prochaine formation complète, Amazon Personalize les `userId` associe à l'historique anonyme des utilisateurs retracé jusqu'à l'original`sessionId`.

Une fois le recyclage terminé, les recommandations seront basées sur l'activité suivie à la fois à `sessionId` partir des événements anonymes et de tout événement suivi en fonction de ceux-ci`userId`.

> **Note**
>
> Si votre utilisateur ne crée pas de compte et que vous souhaitez qu'Amazon Personalize utilise les données lors de l'entraînement, vous pouvez les utiliser `userId` dans le `sessionId` cadre d'événements. Toutefois, si l'utilisateur finit par créer un compte, vous ne pourrez pas associer les événements de sa navigation anonyme à son nouveau compte`userId`.

Services tiers de suivi des événements

Les plateformes de données clients (CDPs) suivantes peuvent vous aider à collecter des données sur les événements depuis votre application et à les envoyer à Amazon Personalize.

- Amplitude — Vous pouvez utiliser Amplitude pour suivre les actions des utilisateurs afin de vous aider à comprendre le comportement de vos utilisateurs. Pour plus d'informations sur l'utilisation d'Amplitude et d'Amazon Personalize, consultez le billet de blog du réseau de AWS partenaires (APN) suivant : Measuring the Effectiveness of Personalization with Amplitude and Amazon Personalize.
- Segment — Vous pouvez utiliser Segment pour envoyer vos données à Amazon Personalize. Pour plus d'informations sur l'intégration de Segment à Amazon Personalize, consultez Amazon Personalize Destination.

Exemples d’implémentations

Pour un exemple de bloc-notes Jupyter expliquant comment utiliser Amazon Personalize pour réagir au comportement en temps réel des utilisateurs à l'aide d'un outil de suivi des événements et de l'PutEventsopération, consultez 2.view_campaign_and_interactions.ipynb dans le dossier getting_started du référentiel. amazon-personalize-samples GitHub

Pour un exemple montrant comment diffuser des événements provenant d'utilisateurs interagissant avec les recommandations, consultez streaming_events dans le référentiel d'exemples Amazon Personalize. GitHub

Pour un exemple complet contenant le code source et les fichiers de support permettant de déployer en temps réel APIs entre vos ressources Amazon Personalize et les applications clientes, consultez

la section Personnalisation en temps réel APIs dans le GitHub référentiel d' AWS exemples. Ce projet explique comment mettre en œuvre les éléments suivants :

- Contexte utilisateur et collecte d'événements utilisateur
- Mise en cache des réponses
- Recommandations de décoration basées sur les métadonnées des articles
- Test A/B
- Authentification par API

Maintien de la pertinence des recommandations

Des recommandations pertinentes peuvent augmenter l'engagement des utilisateurs, le taux de clics et le taux de conversion de votre application à mesure que votre catalogue s'agrandit. Pour maintenir et améliorer la pertinence des recommandations d'Amazon Personalize pour vos utilisateurs, maintenez vos données et vos ressources personnalisées à jour. Cela permet à Amazon Personalize de tirer des enseignements du comportement le plus récent de vos utilisateurs et d'inclure vos derniers articles dans les recommandations.

Rubriques

- Maintien à jour des ensembles de données
- Gestion des recommandeurs de domaines
- Maintien de solutions personnalisées

Maintien à jour des ensembles de données

Au fur et à mesure que votre catalogue s'agrandit, mettez à jour vos données historiques avec des opérations d'importation de données groupées ou individuelles. Pour plus d'informations sur l'importation de données historiques, consultezImportation de données de formation dans les ensembles de données Amazon Personalize. Pour plus d'informations sur la manière dont les données que vous importez après l'entraînement d'un modèle influencent les recommandations, consultezMise à jour des données dans les ensembles de données après l'entraînement.

Pour les cas d'utilisation et les recettes fournissant des recommandations personnalisées en temps réel, maintenez votre jeu de données d'interactions entre les articles à jour en fonction du comportement de vos utilisateurs. Pour ce faire, enregistrez les interactions entre les éléments avec un outil de suivi des événements et le fonctionnement de l' PutEvents API. Amazon Personalize met à jour les recommandations en fonction de l'activité la plus récente de vos utilisateurs lorsqu'ils interagissent avec votre catalogue. Pour plus d'informations sur la personnalisation en temps réel, consultezPersonnalisation en temps réel. Pour plus d'informations sur l'enregistrement d'événements en temps réel, consultezEnregistrement des événements en temps réel pour influencer les recommandations.

Gestion des recommandeurs de domaines

Amazon Personalize reforme automatiquement les modèles qui soutiennent vos recommandations tous les 7 jours. Il s'agit d'un recyclage complet qui crée des modèles entièrement nouveaux basés sur l'intégralité des données de vos ensembles de données. Si vous modifiez les colonnes utilisées lors de la formation, Amazon Personalize lance automatiquement un réentraînement complet des modèles qui soutiennent votre recommandateur.

- Pour les cas d'utilisation les plus intéressants pour vous et les cas d'utilisation recommandés pour vous, Amazon Personalize met à jour votre outil de recommandation pour qu'il prenne en compte les nouveaux articles à recommander. Les mises à jour automatiques ne constituent pas un recyclage complet dans le cadre duquel le modèle apprend du comportement de vos utilisateurs. Les mises à jour automatiques permettent plutôt à Amazon Personalize de présenter vos nouveaux articles dans les recommandations avant la prochaine formation complète du recommandant. Pour plus d'informations sur les mises à jour automatiques, consultezMises à jour automatiques.
- Si vous utilisez le cas d'utilisation Trending now, Amazon Personalize évalue automatiquement les données de vos interactions toutes les deux heures et identifie les articles tendance. Vous n'avez pas à attendre que votre conseiller se reforme.

Pendant que le programme de recyclage est en cours, vous pouvez toujours obtenir des recommandations de la part du recommandant. Jusqu'à ce que le recyclage soit terminé, le recommandeur utilise la configuration et les modèles précédents. Pour suivre les mises à jour, vous pouvez consulter l'horodatage de la dernière mise à jour du recommendeur sur la page de détails du recommendeur de la console Amazon Personalize. Vous pouvez également consulter les `latestRecommenderUpdate` détails de l'DescribeRecommenderopération.

Maintien de solutions personnalisées

Par défaut, toutes les nouvelles solutions utilisent une formation automatique pour créer une nouvelle version de solution tous les 7 jours. L'entraînement se poursuit jusqu'à ce que vous supprimiez la solution.

Lorsque vous créez une solution, nous vous recommandons d'utiliser la formation automatique pour gérer la création des versions de la solution. Cela facilite la maintenance de votre solution. Cela supprime la formation manuelle requise pour que la solution puisse tirer parti de vos données les plus récentes. Sans formation automatique, vous devez créer manuellement de nouvelles versions

de solution pour que la solution puisse tirer des leçons de vos données les plus récentes. Pour plus d'informations sur la configuration de l'entraînement automatique, consultezConfiguration de l'entraînement automatique.

La fréquence de vos formations dépend des besoins de votre entreprise, de la recette que vous utilisez et de la fréquence à laquelle vous importez des données. Pour toutes les recettes, nous recommandons de vous entraîner au moins une fois par semaine. Dans le cas de l'entraînement automatique, il s'agit de la fréquence d'entraînement par défaut. Si vous ajoutez fréquemment de nouveaux objets ou actions, vous souhaiterez peut-être augmenter la fréquence d'entraînement, en fonction de votre recette.

- Si vous utilisez User-Personalization-v 2, Personnalisation utilisateur, ou Next-Best-Action, la solution est automatiquement mise à jour pour prendre en compte les nouveaux éléments ou actions à recommander. Les mises à jour automatiques ne sont pas la même chose que l'entraînement automatique. Une mise à jour automatique ne crée pas une version entièrement nouvelle de la solution, et le modèle ne tire aucune leçon de vos dernières données. Pour maintenir votre solution, votre fréquence d'entraînement doit toujours être d'au moins une fois par semaine. Pour plus d'informations sur les mises à jour automatiques, y compris les directives et exigences supplémentaires, voirMises à jour automatiques.
- Si vous utilisez Trending-Now, Amazon Personalize identifie automatiquement les éléments les plus populaires dans vos données d'interactions sur un intervalle de temps configurable. Trending-Now peut recommander des éléments ajoutés depuis le dernier entraînement par le biais de données d'interactions en masse ou en streaming. Votre fréquence d'entraînement doit toujours être d'au moins une fois par semaine. Pour de plus amples informations, veuillez consulter Recette Trending-Now.
- Si vous n'utilisez pas de recette avec mises à jour automatiques ou la recette Trending-Now, Amazon Personalize prend en compte les nouveaux articles pour les recommandations uniquement après la prochaine formation. Par exemple, si vous utilisez la recette Similar-Items et que vous ajoutez de nouveaux articles tous les jours, vous devez utiliser une fréquence d'entraînement quotidienne pour que ces éléments apparaissent dans les recommandations le jour même.

Mise à jour des données dans les ensembles de données après l'entraînement

À mesure que votre catalogue s'agrandit, importez des données d'entraînement supplémentaires dans vos ensembles de données. Cela permet de maintenir et d'améliorer la pertinence des recommandations d'Amazon Personalize. Vous pouvez importer davantage de données grâce à des opérations d'importation de données groupées ou individuelles.

- Lors des importations individuelles, Amazon Personalize ajoute les nouveaux enregistrements à l'ensemble de données. Pour mettre à jour un élément, un utilisateur ou une action individuel, vous pouvez importer un enregistrement avec le même ID mais avec les attributs modifiés. Vous pouvez importer jusqu'à 10 enregistrements par opération d'importation individuelle.

  Pour plus d'informations sur l'importation individuelle d'enregistrements, consultezImportation d'enregistrements individuels dans un ensemble de données Amazon Personalize. Pour plus d'informations sur l'enregistrement d'événements en temps réel, consultezEnregistrement des événements en temps réel pour influencer les recommandations.
- Dans le cas des importations groupées, vous ajoutez ou remplacez des données en masse en créant une autre tâche d'importation. Par défaut, une tâche d'importation de jeu de données remplace toutes les données existantes du jeu de données que vous avez importé en bloc. Vous pouvez plutôt ajouter les nouveaux enregistrements aux données existantes en modifiant le mode d'importation de la tâche.

  Pour ajouter des données à un jeu de données d'interactions d'éléments ou à un jeu de données d'interactions d'action avec une tâche d'importation de jeu de données, vous devez disposer d'au moins 1 000 nouveaux enregistrements d'interactions entre éléments ou d'interactions d'action. Dans les 20 minutes suivant l'importation en masse, Amazon Personalize met à jour tous les filtres que vous avez créés dans le groupe de jeux de données avec vos nouvelles données en masse. Cette mise à jour permet à Amazon Personalize d'utiliser les données les plus récentes pour filtrer les recommandations destinées à vos utilisateurs.

Après avoir créé un jeu de données Items ou Users, vous pouvez remplacer son schéma par un nouveau ou un schéma existant. Vous pouvez remplacer la structure d'un ensemble de données si votre structure de données a changé après avoir créé le jeu de données. Par exemple, vous pouvez avoir une nouvelle colonne de métadonnées d'articles que vous souhaitez qu'Amazon Personalize prenne en compte pendant la formation. Vous pouvez également ajouter une colonne de données

à utiliser uniquement pour filtrer les recommandations. Pour de plus amples informations, veuillez consulter Remplacement du schéma d'un ensemble de données pour ajouter de nouvelles colonnes.

Une fois que vous avez créé une version de recommandation ou de solution personnalisée, l'influence des nouvelles données sur les recommandations dépend de leur type, de la méthode d'importation et du cas d'utilisation du domaine ou de la recette personnalisée que vous utilisez. Les sections suivantes expliquent comment les nouvelles données influencent les recommandations en temps réel et par lots avant la prochaine formation.

Rubriques

- Comment les nouvelles données influencent les recommandations en temps réel
- Comment les nouvelles données influencent les recommandations par lots (ressources personnalisées)

Comment les nouvelles données influencent les recommandations en temps réel

Une fois que vous avez créé une version de recommandation ou de solution personnalisée, l'influence des nouvelles données sur les recommandations en temps réel dépend du type de données, de la méthode d'importation et du cas d'utilisation du domaine ou de la recette personnalisée que vous utilisez. Les sections suivantes expliquent comment les nouvelles données influencent les recommandations en temps réel avant la prochaine formation.

La formation peut être une formation automatique hebdomadaire dispensée par un recommandant ou une création automatique ou manuelle de versions de solutions. Pour un entraînement manuel avec personnalisation utilisateur, omettez-le `trainingMode` pour utiliser le mode d'entraînement par défaut`FULL`.

Rubriques

- Nouvelles interactions
- Nouveaux articles
- Nouveaux utilisateurs
- Nouvelles actions

Nouvelles interactions

Les nouvelles interactions sont des interactions entre des objets ou des actions que vous importez après le dernier entraînement. Qu'il s'agisse de données en temps réel ou en masse, si les interactions impliquent un nouvel élément ou une nouvelle action, Amazon Personalize peut envisager de le faire pour des recommandations sans formation si votre recette ou votre cas d'utilisation inclut l'exploration. Pour plus d’informations, consultez Nouveaux articles ou Nouvelles actions.

Événements en temps réel

Pour les cas d'utilisation et les recettes proposant une personnalisation en temps réel, Amazon Personalize utilise immédiatement les interactions en temps réel entre un utilisateur et les éléments ou actions présentés lors de la dernière formation. Lors de la génération de recommandations pour l'utilisateur participant à l'événement, Amazon Personalize utilise ces interactions en temps réel. Pour plus d'informations sur la personnalisation en temps réel, consultezPersonnalisation en temps réel.

Pour tous les cas d'utilisation de domaines et les recettes personnalisées qui ne proposent pas de personnalisation en temps réel, comme la recommandation d'éléments similaires, votre modèle apprend des données d'interactions en temps réel uniquement après l'entraînement.

Interactions en masse

Pour les interactions en masse, qu'il s'agisse de tâches d'importation incrémentielle ou complète de jeux de données, votre modèle apprend des interactions par éléments ou des interactions par action en masse uniquement après l'entraînement suivant. Les données en masse ne sont pas utilisées pour mettre à jour les recommandations de personnalisation en temps réel.

Pour plus d'informations sur l'importation d'un plus grand nombre de données en masse, consultezImportation de données en masse dans Amazon Personalize à l'aide d'une tâche d'importation de jeux de données.

Nouveaux articles

Les nouveaux objets sont des objets que vous importez après le dernier entraînement. Ils peuvent provenir de données d'interactions ou de métadonnées d'éléments d'un jeu de données d'articles.

Les nouveaux éléments sont pris en compte pour les recommandations comme suit :

- Pour les domaines Top Sicks for you et Recommended for you, ou User-Personalization-v 2, User-Personalization, ou Next-Best-Action Recipes, Amazon Personalize met automatiquement

à jour le modèle toutes les deux heures. Après chaque mise à jour, Amazon Personalize prend en compte les nouveaux articles pour obtenir des recommandations dans le cadre de l'exploration. Lors de l'examen du nouvel article, Amazon Personalize prend en compte toutes les métadonnées associées à l'article. Toutefois, ces données n'auront un effet plus important sur les recommandations qu'une fois que vous aurez enregistré les interactions relatives à l'élément et que vous aurez entraîné un nouveau modèle. Pour plus d'informations sur les mises à jour, consultezMises à jour automatiques.

- Si vous utilisez le cas d'utilisation Trending now, Amazon Personalize évalue automatiquement les données de vos interactions toutes les deux heures et identifie les articles tendance. Vous n'avez pas à attendre que votre conseiller s'entraîne. Si vous utilisez la recette Trending-Now, Amazon Personalize prend automatiquement en compte tous les nouveaux articles selon des intervalles configurables, sans formation. Pour plus d'informations sur la configuration des intervalles, consultezRecette Trending-Now.
- Si vous n'utilisez pas la recette Trending-Now ou si votre cas d'utilisation ou recette ne prend pas en charge les mises à jour automatiques, Amazon Personalize ne prendra en compte les nouveaux articles qu'après la prochaine formation.

Nouveaux utilisateurs

Les nouveaux utilisateurs sont des utilisateurs que vous importez après la dernière formation. Ils peuvent provenir de données d'interactions ou de métadonnées utilisateur dans un ensemble de données utilisateurs. Pour les nouveaux utilisateurs anonymes (utilisateurs sans UserID), vous pouvez enregistrer les événements de l'utilisateur avec un identifiant `sessionId` et Amazon Personalize associera les événements à l'utilisateur avant qu'il ne se connecte. Pour de plus amples informations, veuillez consulter Enregistrement d'événements pour les utilisateurs anonymes.

Amazon Personalize génère des recommandations pour les nouveaux utilisateurs comme suit :

- Si vous utilisez le cas d'utilisation du domaine Trending now ou la recette personnalisée Trending-Now, les nouveaux utilisateurs reçoivent immédiatement des recommandations pour les articles les plus populaires. Si vous utilisez la recette Popularity-Count, les nouveaux utilisateurs reçoivent immédiatement des recommandations pour les articles présentant le plus d'interactions.
- Pour les recettes ou les cas d'utilisation fournissant des recommandations personnalisées aux utilisateurs, les recommandations destinées aux nouveaux utilisateurs sont basées sur l'historique des premières interactions de vos utilisateurs existants. Les premiers éléments ou actions avec lesquels ces utilisateurs existants ont interagi sont plus susceptibles d'être recommandés

aux nouveaux utilisateurs. Pour les recettes de personnalisation utilisateur ou de classement personnalisé, si vous définissez `recency_mask` cette option, les recommandations incluent également des éléments basés sur les dernières tendances de popularité dans vos données d'interactions. `true`

Les éléments suivants peuvent accroître la pertinence des recommandations pour les nouveaux utilisateurs :

- Données relatives aux interactions : le principal moyen d'améliorer la pertinence des recommandations pour un nouvel utilisateur est d'importer les données issues de ses interactions avec vos articles. Pour plus d'informations sur la manière dont les nouvelles données d'interactions influencent les recommandations, voirNouvelles interactions.
- Métadonnées utilisateur : l'importation de métadonnées utilisateur, telles que GENDER ou MEMBERSHIP_STATUS, peut améliorer les recommandations. Pour que les métadonnées influencent les recommandations, vous devez attendre la fin de la formation automatique hebdomadaire de votre recommandeur de domaine. Vous devez également créer manuellement une nouvelle version de solution.
- Métadonnées contextuelles : si votre cas d'utilisation ou votre recette prend en charge les métadonnées contextuelles et que votre jeu de données sur les interactions entre articles contient des champs de métadonnées pour les données contextuelles, vous pouvez fournir le contexte de l'utilisateur dans votre demande de recommandations. Cela ne nécessite pas de reconversion. Pour de plus amples informations, veuillez consulter Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles.

Nouvelles actions

Les nouvelles actions sont des actions que vous importez depuis la dernière formation. Ils peuvent provenir de données d'interaction d'action ou d'actions d'un ensemble de données Actions.

Avec Next-Best-Action cette recette, Amazon Personalize met automatiquement à jour une version de la solution toutes les deux heures. Après chaque mise à jour, Amazon Personalize envisage de nouvelles actions et recommandations dans le cadre de l'exploration. Lors de l'examen de la nouvelle action, Amazon Personalize prend en compte toutes les métadonnées associées à l'action. Toutefois, ces données n'auront un effet plus important sur les recommandations qu'une fois que vous aurez enregistré les interactions liées à l'action et que vous vous serez complètement réentraîné. Pour plus d'informations sur les mises à jour, voir Mises à jour automatiques

Comment les nouvelles données influencent les recommandations par lots (ressources personnalisées)

Après avoir créé une version de solution personnalisée, l'influence des nouvelles données sur les recommandations par lots dépend du type de données, de la méthode d'importation et de la recette personnalisée que vous utilisez.

Pour les segments d'utilisateurs, Amazon Personalize génère des segments en utilisant uniquement les données présentes lors de la dernière formation sur la version complète de la solution. Et Amazon Personalize utilise uniquement les données en masse que vous avez importées en mode d'importation COMPLET (en remplacement des données existantes). Pour plus d'informations sur les segments d'utilisateurs, consultezObtenir des segments d'utilisateurs par lots avec des ressources personnalisées.

Lors de la génération des recommandations relatives aux articles par lots, Amazon Personalize prend en compte toutes les données groupées présentes au moment de la création de la dernière version de la solution. Ces données peuvent être importées avec le mode d'importation FULL ou INCREMENTAL. Pour que les nouveaux enregistrements groupés influencent les recommandations par lots, vous devez créer une nouvelle version de solution, puis créer la tâche d'inférence par lots.

Les sections suivantes expliquent comment les importations individuelles influencent les recommandations relatives aux articles par lots.

Rubriques

- Nouvelles interactions
- Nouveaux utilisateurs
- Nouveaux articles

Nouvelles interactions

Si vous utilisez une recette USER_PERSONALIZATION ou PERSONALIZED_RANKING, Amazon Personalize prend en compte les données relatives aux interactions entre les articles existants et les utilisateurs dans les 15 minutes suivant l'importation des données. Ces éléments et utilisateurs doivent avoir été présents lors de la dernière formation. Pour vous assurer que les événements sont pris en compte, nous vous recommandons d'attendre au moins 15 minutes avant de démarrer une tâche d'inférence par lots. Pour toutes les autres recettes et pour les événements impliquant de

nouveaux articles ou de nouveaux utilisateurs, vous devez créer une nouvelle version de solution pour les événements diffusés en continu afin d'influencer les recommandations par lots.

Nouveaux utilisateurs

Pour les utilisateurs ne disposant pas de données d'interactions, les recommandations concernent initialement uniquement les éléments populaires. Si vous utilisez une recette USER_PERSONALIZATION ou PERSONALIZED_RANKING et que vous enregistrez des événements pour l'utilisateur, ses recommandations peuvent devenir plus pertinentes environ 15 minutes après l'importation, sans formation préalable. Pour vous assurer que les événements sont pris en compte, nous vous recommandons d'attendre au moins 15 minutes avant de démarrer une tâche d'inférence par lots. Pour toutes les autres recettes, vous devez créer une nouvelle version de solution pour les événements diffusés en continu afin d'influencer les recommandations par lots pour les utilisateurs ne disposant pas de données d'interaction.

Nouveaux articles

Avec User-Personalization-v 2 et User-Personalization, lorsque vous créez une tâche d'inférence par lots et que vous spécifiez la dernière version de solution entièrement entraînée pour votre solution, Amazon Personalize met automatiquement à jour la version de la solution pour inclure de nouveaux éléments dans les recommandations avec exploration. Si vous ne spécifiez pas la dernière version de la solution, aucune mise à jour n'est effectuée. Pour toute autre recette, vous devez créer une nouvelle version de solution pour que les nouveaux éléments figurent dans les recommandations par lots. Pour plus d'informations sur l'exploration, consultezExploration.

Remplacement du schéma d'un ensemble de données pour ajouter de nouvelles colonnes

Après avoir créé un jeu de données Items ou Users, vous pouvez remplacer son schéma par un nouveau ou un schéma existant. Vous pouvez remplacer la structure d'un ensemble de données si votre structure de données a changé après avoir créé le jeu de données. Par exemple, vous pouvez avoir une nouvelle colonne de métadonnées d'articles que vous souhaitez qu'Amazon Personalize prenne en compte pendant la formation. Vous pouvez également ajouter une colonne de données à utiliser uniquement pour filtrer les recommandations.

Lorsque vous remplacez le schéma d'un jeu de données, vous devez conserver tous les champs du schéma précédent et vous ne pouvez pas modifier leurs types de données ou leurs attributs. Une fois que vous avez remplacé le schéma d'un ensemble de données, Amazon Personalize exclut automatiquement toute nouvelle colonne de la formation des recommandateurs ou des solutions personnalisées existants. Pour plus de directives et d'exigences, voirDirectives et exigences.

Vous pouvez remplacer le schéma d'un ensemble de données par la console Amazon Personalize AWS Command Line Interface (AWS CLI) et AWS les SDK.

Rubriques

- Directives et exigences
- Remplacement du schéma d'un ensemble de données (console)
- Remplacement du schéma d'un ensemble de données (AWS CLI)
- Remplacement du schéma d'un ensemble de données (AWS SDK)

Directives et exigences

Avant de remplacer le schéma d'un ensemble de données, assurez-vous de connaître les directives et exigences suivantes :

- Vous ne pouvez pas remplacer le schéma d'un jeu de données d'interactions Item, d'un jeu de données d'interactions Action ou d'un jeu de données Actions.
- Vous pouvez ajouter de nouveaux champs à votre schéma de remplacement, mais vous devez conserver tous les champs du schéma précédent. Et vous ne pouvez pas modifier leurs types de données ou leurs attributs. Par exemple, si le schéma précédent inclut un `MEMBERSHIP_STATUS`

champ pour les données de chaîne catégorielles, le nouveau schéma que vous utilisez doit inclure un `MEMBERSHIP_STATUS` champ avec ces attributs et types de données.

- Si le schéma actuel comporte un champ que vous souhaitez renommer, ou si vous souhaitez modifier ses types de données ou ses attributs, vous pouvez ajouter un nouveau champ avec un nouveau nom et des types ou attributs modifiés. Incluez ensuite le nouveau champ dans la formation et excluez l'ancien champ. Tout nouveau champ doit contenir des `null` données. Si l'ancien champ ne prenait pas en charge les données nulles, lorsque vous importez des données, vous pouvez utiliser des données de substitution pour vous assurer que votre importation correspond au schéma. Pour plus d'informations sur la configuration des colonnes utilisées par un recommandeur, consultezMettre à jour un recommandeur. Pour plus d'informations sur la configuration des colonnes utilisées par une solution, consultezConfiguration des colonnes utilisées lors de l'entraînement.
- Tout nouveau champ doit contenir des `null` données. Pour plus d'informations sur l'ajout d'un type nul à un champ, consultezTypes de données de schéma.
- Une fois que vous avez remplacé le schéma d'un ensemble de données, Amazon Personalize exclut automatiquement toute nouvelle colonne de la formation des recommandateurs ou des solutions personnalisées existants. L'utilisation du jeu de données modifié implique les actions suivantes :
  - Pour utiliser de nouvelles colonnes lors de l'entraînement, importez des données conformes au nouveau schéma. Mettez ensuite à jour tous les recommandateurs pour qu'ils utilisent de nouvelles colonnes, ou créez une nouvelle solution personnalisée et configurez les colonnes qu'elle utilise lors de l'entraînement.

    Pour plus d'informations sur la mise à jour des colonnes utilisées par un recommandeur, consultezMettre à jour un recommandeur. Pour plus d'informations sur la configuration des colonnes utilisées par une solution, consultezConfiguration des colonnes utilisées lors de l'entraînement.
  - Pour utiliser des colonnes uniquement lors du filtrage, importez des données conformes au nouveau schéma, créez un filtre qui utilise les nouvelles données et appliquez votre filtre à vos demandes de recommandation. Vous n'avez pas besoin de mettre à jour les recommandations, ni de créer ou de mettre à jour de ressources personnalisées.

Remplacement du schéma d'un ensemble de données (console)

Pour remplacer le schéma d'un ensemble de données par la console Amazon Personalize, vous choisissez le jeu de données à modifier et choisissez de le remplacer par un nouveau schéma ou d'utiliser un schéma existant.

Pour remplacer le schéma d'un ensemble de données

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, choisissez Datasets, puis cliquez sur le bouton radio correspondant au jeu de données que vous souhaitez modifier.
4. Choisissez Actions, puis sélectionnez Remplacer le schéma.
5. Dans Détails du schéma, choisissez de le remplacer par un nouveau schéma ou un schéma créé précédemment.
6. Spécifiez le nouveau schéma à utiliser. Si vous avez choisi de :
   - Remplacez-le par un nouveau schéma, puis nommez-le et, dans Définition du schéma, apportez vos modifications au schéma JSON.
   - Utilisez un schéma créé précédemment, puis dans Schéma créé précédemment, choisissez le schéma que vous souhaitez utiliser. Seuls les schémas éligibles sont répertoriés. Pour plus d'informations sur les exigences relatives aux schémas, consultezDirectives et exigences.
7. Choisissez Remplacer. Lorsque le jeu de données est actif, vous pouvez commencer à importer des données conformes au nouveau schéma. Pour plus d'informations, consultez Importation de données de formation dans les ensembles de données Amazon Personalize.

Remplacement du schéma d'un ensemble de données (AWS CLI)

Pour remplacer le schéma d'un ensemble de données par le AWS CLI, utilisez la `update-dataset` commande, spécifiez le nom de ressource Amazon (ARN) de l'ensemble de données à mettre à jour et l'ARN du nouveau schéma à utiliser. Vous ne pouvez pas mettre à jour le schéma d'un jeu de données d'interactions Item, d'un jeu de données d'interactions Action ou d'un jeu de données Actions.

Le code suivant montre comment mettre à jour le schéma d'un ensemble de données avec le AWS CLI. Pour remplacer le schéma d'un ensemble de données par un nouveau, utilisez d'abord la `create-schema` commande. Utilisez ensuite le code suivant pour remplacer le schéma actuel par le nouveau. Pour plus d'informations sur la création d'un schéma à l'aide du AWS CLI, consultezCréation d'un jeu de données et d'un schéma (AWS CLI). Pour plus d'informations sur les ensembles de données et les exigences en matière de schéma, consultezCréation de fichiers JSON pour les schémas Amazon Personalize.

```
aws personalize update-dataset \
--dataset-arn Dataset ARN \
--schema-arn New schema ARN
```

Lorsque le jeu de données est actif, vous pouvez commencer à importer des données conformes au nouveau schéma. Pour plus d'informations, consultez Importation de données de formation dans les ensembles de données Amazon Personalize. Pour plus d'informations sur la dernière mise à jour du jeu de données, vous pouvez utiliser l'DescribeDatasetopération.

Remplacement du schéma d'un ensemble de données (AWS SDK)

Pour remplacer le schéma d'un ensemble de données par les AWS SDK, vous devez utiliser l'opération `UpdateDataset` API. Spécifiez le nom de ressource Amazon (ARN) de l'ensemble de données à mettre à jour et le nouveau schéma à utiliser. Vous ne pouvez pas mettre à jour le schéma d'un jeu de données d'interactions Item, d'un jeu de données d'interactions Action ou d'un jeu de données Actions.

Le code suivant montre comment remplacer le schéma d'un ensemble de données par le AWS SDK pour Python (Boto3). Pour remplacer le schéma d'un ensemble de données par un nouveau, utilisez d'abord l'CreateSchemaopération. Utilisez ensuite le code suivant pour remplacer le schéma actuel par le nouveau. Pour plus d'informations sur la création d'un schéma à l'aide AWS des SDK, consultezCréation d'un jeu de données et d'un schéma (AWS SDKs). Pour plus d'informations sur les exigences relatives aux ensembles de données et aux schémas, consultezCréation de fichiers JSON pour les schémas Amazon Personalize.

```
import boto3

personalize = boto3.client('personalize')

update_dataset_response = personalize.update_dataset(
```

```
    datasetArn = 'dataset_arn',
    schemaArn = 'new_schema_arn'
)

print(update_dataset_response)
```

Lorsque le jeu de données est actif, vous pouvez commencer à importer des données conformes au nouveau schéma. Pour plus d'informations, consultez Importation de données de formation dans les ensembles de données Amazon Personalize. Pour plus d'informations sur la dernière mise à jour du jeu de données, vous pouvez utiliser l'DescribeDatasetopération.

Exportation des données d'entraînement d'un ensemble de données vers Amazon S3

Après avoir importé vos données dans un ensemble de données Amazon Personalize, vous pouvez les exporter vers un compartiment Amazon S3. Vous pouvez exporter des données pour vérifier et inspecter les données utilisées par Amazon Personalize pour générer des recommandations, consulter les événements d'interaction avec les articles que vous avez précédemment enregistrés en temps réel ou effectuer une analyse hors ligne de vos données.

Vous pouvez choisir d'exporter uniquement les données que vous avez importées en masse (importées à l'aide d'une tâche d'importation de jeux de données Amazon Personalize), uniquement les données que vous avez importées individuellement (enregistrements importés à l'aide de la console ou `PutItems` des opérations `PutEventsPutUsers`, ou), ou les deux.

> **Note**
>
> Vous ne pouvez pas exporter de données dans un jeu de données d'interactions Action ou dans un jeu de données d'actions.

Pour les enregistrements qui correspondent exactement à tous les champs, Amazon Personalize n'exporte qu'un seul enregistrement. Si deux enregistrements ont le même identifiant mais qu'un ou plusieurs champs sont différents, Amazon Personalize inclut ou supprime les enregistrements en fonction des données que vous choisissez d'exporter :

- Si vous exportez des données en masse et incrémentielles, Amazon Personalize exporte uniquement les éléments les plus récents avec le même identifiant (dans les exportations de jeux de données d'articles), et uniquement les utilisateurs ayant le même identifiant (dans les exportations de jeux de données utilisateurs). Pour les ensembles de données sur les interactions entre articles, Amazon Personalize exporte toutes les données relatives aux interactions entre articles.
- Si vous exportez uniquement des données incrémentielles, Amazon Personalize exporte toutes les données relatives aux articles, aux utilisateurs ou aux interactions entre articles que vous avez importées individuellement, y compris les articles ou les utilisateurs ayant les mêmes identifiants. Seuls les enregistrements correspondant exactement à tous les champs sont exclus.

- Si vous exportez uniquement des données en masse, Amazon Personalize inclut toutes les données relatives aux articles, aux utilisateurs ou aux interactions entre articles que vous avez importées en masse, y compris les articles ou les utilisateurs ayant les mêmes identifiants. Seuls les enregistrements correspondant exactement à tous les champs sont exclus.

Pour exporter un ensemble de données, vous devez créer une tâche d'exportation de jeu de données. Une tâche d'exportation d'ensembles de données est un outil d'exportation d'enregistrements qui génère les enregistrements d'un ensemble de données vers un ou plusieurs fichiers CSV d'un compartiment Amazon S3. Le fichier CSV en sortie inclut une ligne d'en-tête dont les noms de colonnes correspondent aux champs du schéma du jeu de données.

Rubriques

- Exigences relatives aux autorisations d'exportation de jeux de données
- Création d'une tâche d'exportation de jeux de données dans Amazon Personalize

Exigences relatives aux autorisations d'exportation de jeux de données

Pour exporter un ensemble de données, Amazon Personalize doit être autorisé à ajouter des fichiers à votre compartiment Amazon S3. Pour accorder des autorisations, associez une nouvelle politique Gestion des identités et des accès AWS (IAM) à votre rôle de service Amazon Personalize qui accorde au rôle l'autorisation d'utiliser les `ListBucket` actions `PutObject` et sur votre compartiment, et attachez une politique de compartiment à votre compartiment Amazon S3 de sortie qui accorde à Amazon Personalize l'autorisation principale d'utiliser les `ListBucket` actions `PutObject` et.

Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Politique de rôle de service pour l'exportation d'un ensemble de données

L'exemple de politique suivant accorde à votre rôle de service Amazon Personalize l'autorisation d'utiliser les `ListBucket` actions `PutObject` et. `amzn-s3-demo-bucket`Remplacez-le par le nom de votre bucket de sortie. Pour plus d'informations sur l'attachement de politiques à un rôle de service IAM, consultezAssocier une politique Amazon S3 à votre rôle de service Amazon Personalize.

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Politique relative aux compartiments Amazon S3 pour l'exportation d'un ensemble de données

L'exemple de politique suivant accorde à Amazon Personalize l'autorisation d'utiliser `ListBucket` les actions `PutObject` et sur un compartiment Amazon S3. Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment. Pour plus d'informations sur l'ajout d'une politique de compartiment Amazon S3 à un compartiment, consultez Ajouter une politique de compartiment à l'aide de la console Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Création d'une tâche d'exportation de jeux de données dans Amazon Personalize

Vous pouvez créer une tâche d'exportation de jeux de données à l'aide de la console Amazon Personalize AWS Command Line Interface (AWS CLI) ou AWS des SDK.

Création d'une tâche d'exportation de jeux de données (console)

Après avoir importé vos données dans un ensemble de données et créé un compartiment Amazon S3 en sortie, vous pouvez exporter les données vers le compartiment à des fins d'analyse. Pour exporter un ensemble de données à l'aide de la console Amazon Personalize, vous devez créer une tâche d'exportation de jeu de données. Pour plus d'informations sur la création d'un compartiment Amazon S3, consultez la section Création d'un compartiment dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Avant d'exporter un ensemble de données, assurez-vous que votre rôle de service Amazon Personalize peut accéder à votre compartiment Amazon S3 de sortie et y écrire. Consultez Exigences relatives aux autorisations d'exportation de jeux de données.

Pour créer une tâche d'exportation d'un ensemble de données (console)

1. Ouvrez la console Amazon Personalize à l'adresse https://console.aws.amazon.com/personalize/home.
2. Dans le volet de navigation, sélectionnez Groupes de jeux de données.
3. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.
4. Dans le volet de navigation, sélectionnez Datasets.
5. Choisissez l'ensemble de données que vous souhaitez exporter vers un compartiment Amazon S3.

6. Dans Tâches d'exportation de jeux de données, choisissez Créer une tâche d'exportation de jeu de données.
7. Dans Détails de la tâche d'exportation de l'ensemble de données, pour le nom de la tâche d'exportation de l'ensemble de données, entrez le nom de la tâche d'exportation.
8. Pour le rôle de service IAM, choisissez le rôle de service Amazon Personalize que vous avez créé dansCréation d'un rôle IAM pour Amazon Personalize.
9. Pour le chemin de sortie des données Amazon S3, entrez le compartiment Amazon S3 de destination. Utilisez la syntaxe suivante :

   ```
   s3://amzn-s3-demo-bucket/<folder path>
   ```

10. Si vous utilisez AWS KMS pour le chiffrement, pour l'ARN de la clé KMS, entrez le nom de ressource Amazon (ARN) de la AWS KMS clé.
11. Pour Exporter le type de données, choisissez le type de données à exporter en fonction de la façon dont vous avez initialement importé les données.

    - Choisissez Bulk pour exporter uniquement les données que vous avez importées en bloc à l'aide d'une tâche d'importation de jeux de données.
    - Choisissez Incremental pour exporter uniquement les données que vous avez importées individuellement à l'aide de la console ou `PutItems` des opérations `PutEventsPutUsers`, ou.
    - Choisissez Les deux pour exporter toutes les données de l'ensemble de données.
12. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
13. Choisissez Créer une tâche d'exportation de jeux de données.

    Sur la page d'aperçu de l'ensemble de données, dans Tâches d'exportation de l'ensemble de données, la tâche est répertoriée avec un statut de tâche d'exportation. La tâche d'exportation du jeu de données est terminée lorsque le statut est ACTIF. Vous pouvez ensuite télécharger les données depuis le compartiment Amazon S3 de sortie. Pour plus d'informations sur le téléchargement d'objets depuis un compartiment Amazon S3, consultez la section Téléchargement d'un objet dans le guide de l'utilisateur d'Amazon Simple Storage Service. .

Création d'une tâche d'exportation de jeux de données (AWS CLI)

Après avoir importé vos données dans le jeu de données et créé un compartiment Amazon S3 en sortie, vous pouvez exporter l'ensemble de données vers le compartiment à des fins d'analyse. Pour exporter un ensemble de données à l'aide de AWS CLI, créez une tâche d'exportation d'ensemble de données à l'aide de la `create-dataset-export-job` AWS CLI commande. Pour plus d'informations sur la création d'un compartiment Amazon S3, consultez la section Création d'un compartiment dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Avant d'exporter un ensemble de données, assurez-vous que le rôle de service Amazon Personalize peut accéder à votre compartiment Amazon S3 de sortie et y écrire. Consultez Exigences relatives aux autorisations d'exportation de jeux de données.

Voici un exemple de la commande `create-dataset-export-job` AWS CLI . Donnez un nom à la tâche, remplacez-le `dataset arn` par le nom de ressource Amazon (ARN) de l'ensemble de données que vous souhaitez exporter et remplacez-le `role ARN` par l'ARN du rôle de service Amazon Personalize dans lequel vous l'avez crééCréation d'un rôle IAM pour Amazon Personalize. Dans`s3DataDestination`, pour le`kmsKeyArn`, indiquez éventuellement l'ARN de votre AWS KMS clé, et pour le, `path` fournissez le chemin d'accès à votre compartiment Amazon S3 de sortie.

Pour`ingestion-mode`, spécifiez les données à exporter à partir des options suivantes :

- Spécifiez `BULK` de n'exporter que les données que vous avez importées en masse à l'aide d'une tâche d'importation de jeux de données.
- Spécifiez `PUT` de n'exporter que les données que vous avez importées individuellement à l'aide de la console ou `PutItems` des opérations `PutEvents` PutUsers, ou.
- Spécifiez `ALL` d'exporter toutes les données de l'ensemble de données.

Pour plus d’informations, consultez CreateDatasetExportJob.

```
aws personalize create-dataset-export-job \
  --job-name job name \
  --dataset-arn dataset ARN \
  --job-output "{\"s3DataDestination\":{\"kmsKeyArn\":\"kms key ARN\",\"path\":
\"s3://amzn-s3-demo-bucket/folder-name/\"}}" \
  --role-arn role ARN \
  --ingestion-mode PUT
```

L'ARN de la tâche d'exportation du jeu de données s'affiche.

```
{
  "datasetExportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-export-job/
DatasetExportJobName"
}
```

Utilisez cette `DescribeDatasetExportJob` opération pour vérifier l'état.

```
aws personalize describe-dataset-export-job \
  --dataset-export-job-arn dataset export job ARN
```

Création d'une tâche d'exportation de jeux de données (AWS SDK)

Après avoir importé vos données dans le jeu de données et créé un compartiment Amazon S3 en sortie, vous pouvez exporter l'ensemble de données vers le compartiment à des fins d'analyse. Pour exporter un ensemble de données à l'aide des AWS SDK, créez une tâche d'exportation d'ensemble de données à l'aide de l'CreateDatasetExportJobopération. Pour plus d'informations sur la création d'un compartiment Amazon S3, consultez la section Création d'un compartiment dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Le code suivant montre comment créer une tâche d'exportation de jeu de données à l'aide du SDK AWS SDK pour Python (Boto3) ou du SDK SDK pour Java 2.x.

Avant d'exporter un ensemble de données, assurez-vous que le rôle de service Amazon Personalize peut accéder à votre compartiment Amazon S3 de sortie et y écrire. Consultez Exigences relatives aux autorisations d'exportation de jeux de données.

SDK for Python (Boto3)

Utilisez ce qui suit `create_dataset_export_job` pour exporter les données d'un ensemble de données vers un compartiment Amazon S3. Donnez un nom à la tâche, remplacez-le `dataset arn` par le nom de ressource Amazon (ARN) de l'ensemble de données que vous souhaitez exporter et remplacez-le `role ARN` par l'ARN du rôle de service Amazon Personalize dans lequel vous l'avez crééCréation d'un rôle IAM pour Amazon Personalize. Dans`s3DataDestination`, pour le`kmsKeyArn`, indiquez éventuellement l'ARN de votre AWS KMS clé, et pour le, `path` fournissez le chemin d'accès à votre compartiment Amazon S3 de sortie.

Pour`ingestionMode`, spécifiez les données à exporter à partir des options suivantes :

- Spécifiez `BULK` de n'exporter que les données que vous avez importées en masse à l'aide d'une tâche d'importation de jeux de données.

- Spécifiez PUT de n'exporter que les données que vous avez importées individuellement à l'aide de la console ou `PutItems` des opérations `PutEvents` PutUsers, ou.
- Spécifiez `ALL` d'exporter toutes les données de l'ensemble de données.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_export_job(
    jobName = 'job name',
    datasetArn = 'dataset ARN',
    jobOutput = {
      "s3DataDestination": {
        "kmsKeyArn": "kms key ARN",
        "path": "s3://amzn-s3-demo-bucket/folder-name/"
      }
    },
    roleArn = 'role ARN',
    ingestionMode = 'PUT'
)

dsej_arn = response['datasetExportJobArn']

print ('Dataset Export Job arn: ' + dsej_arn)

description = personalize.describe_dataset_export_job(
    datasetExportJobArn = dsej_arn)['datasetExportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetExportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

Utilisez la `createDatasetExportJob` méthode suivante pour créer une tâche d'exportation de jeu de données. Transmettez les paramètres suivants : a PersonalizeClient, le nom de votre tâche d'exportation, l'ARN du jeu de données que vous souhaitez exporter, le mode d'ingestion, le chemin du compartiment Amazon S3 de sortie et l'ARN de votre AWS KMS clé.

Il `ingestionMode` peut s'agir de l'une des options suivantes :

- `IngestionMode.BULK`À utiliser pour exporter uniquement les données que vous avez importées en masse à l'aide d'une tâche d'importation de jeux de données.
- `IngestionMode.PUT`À utiliser pour exporter uniquement les données que vous avez importées individuellement à l'aide de la console ou `PutItems` des opérations `PutEvents` PutUsers, ou.
- Permet `IngestionMode.ALL` d'exporter toutes les données de l'ensemble de données.

```
public static void createDatasetExportJob(PersonalizeClient personalizeClient,
                                          String jobName,
                                          String datasetArn,
                                          IngestionMode ingestionMode,
                                          String roleArn,
                                          String s3BucketPath,
                                          String kmsKeyArn) {

    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String status = null;

    try {
        S3DataConfig exportS3DataConfig = S3DataConfig.builder()
            .path(s3BucketPath)
            .kmsKeyArn(kmsKeyArn)
            .build();

        DatasetExportJobOutput jobOutput = DatasetExportJobOutput.builder()
            .s3DataDestination(exportS3DataConfig)
            .build();

        CreateDatasetExportJobRequest createRequest =
 CreateDatasetExportJobRequest.builder()
            .jobName(jobName)
            .datasetArn(datasetArn)
            .ingestionMode(ingestionMode)
            .jobOutput(jobOutput)
            .roleArn(roleArn)
            .build();

        String datasetExportJobArn =
 personalizeClient.createDatasetExportJob(createRequest).datasetExportJobArn();
```

```
        DescribeDatasetExportJobRequest describeDatasetExportJobRequest =
 DescribeDatasetExportJobRequest.builder()
            .datasetExportJobArn(datasetExportJobArn)
            .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetExportJob datasetExportJob =
 personalizeClient.describeDatasetExportJob(describeDatasetExportJobRequest)
                .datasetExportJob();

            status = datasetExportJob.status();
            System.out.println("Export job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Conditions requises pour supprimer des ressources Amazon Personalize

La suppression de ressources peut vous aider à éviter des coûts inutiles. Par exemple, vous devez payer des frais de campagne lorsqu'une campagne est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour obtenir la liste complète des frais et des prix, consultez la section Tarification Amazon Personalize.

Pour supprimer des ressources à l'aide de la console Amazon Personalize, sélectionnez Supprimer sur la page de détails de la ressource. Pour supprimer une ressource avec Amazon Personalize APIs, vous devez utiliser le `Delete` APIs avec le SDKs ou le AWS Command Line Interface (AWS CLI).

Pour connaître les étapes détaillées relatives à la suppression d'un ensemble de données à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) AWS SDKs, ou consultezSupprimer un ensemble de données pour supprimer toutes ses données. Vous pouvez appliquer les modèles décrits dans ces étapes à d'autres ressources Amazon Personalize. Pour plus d'informations sur la suppression d'utilisateurs et de leurs données dans votre groupe de jeux de données, consultezSupprimer des utilisateurs et leurs données à l'aide d'une tâche de suppression de données.

Vous devez supprimer certaines ressources avant d'en supprimer d'autres. Par exemple, si vous créez un outil de suivi des événements et un jeu de données d'interactions entre les éléments, vous devez supprimer le suivi des événements avant de pouvoir supprimer le jeu de données. Les sections suivantes fournissent des directives et des exigences de commande relatives à la suppression des ressources Amazon Personalize.

Rubriques

- Directives relatives à la suppression de ressources
- Ordre recommandé pour la suppression des ressources
- Supprimer des utilisateurs et leurs données à l'aide d'une tâche de suppression de données
- Supprimer un ensemble de données pour supprimer toutes ses données

Directives relatives à la suppression de ressources

Voici les instructions relatives à la suppression de ressources :

- La suppression d'une ressource dans Amazon Personalize est une action irréversible. La suppression ne peut pas être arrêtée une fois qu'elle a commencé.
- Vous ne pouvez pas supprimer une ressource dont le statut passe d'un état à un autre. Par exemple, vous ne pouvez pas supprimer une ressource créée en attente ou en cours. Le statut de la ressource doit être ACTIVE (ACTIF) ou CREATE FAILED (ÉCHEC DE LA CRÉATION). Cela inclut le `latestSolutionUpdate` statut d'une solution. Vous pouvez vérifier l'état d'une ressource à l'aide du `Describe` APIs. Par exemple, le fonctionnement de DescribeCampaign l'API.
- Pour plus d'informations sur la suppression des données d'entraînement dans Amazon S3, consultez Comment supprimer des objets d'un compartiment S3 ? .
- Les tâches d'importation de jeux de données ne vous sont pas facturées une fois qu'elles sont terminées et vous ne pouvez pas les supprimer.
- Les schémas ne vous sont pas facturés et vous ne pouvez pas supprimer un schéma avec la console Amazon Personalize. Pour supprimer un schéma, utilisez l'opération DeleteSchema API.

Les exigences spécifiques à la suppression d'ensembles de données sont les suivantes :

- Vous devez supprimer tous les filtres avant de supprimer un ensemble de données.
- Si vous avez créé un outil de suivi des événements, vous devez le supprimer avant de supprimer un ensemble de données d'interactions entre les éléments.
- Si vous avez créé une attribution métrique qui fait référence au jeu de données, vous devez d'abord supprimer l'attribution métrique.
- Si vous utilisez User-Personalization-v 2 scénarios d'utilisation (Personnalisation utilisateur) ou Next-Best-Action recettes ou Top picks for you et Recommended for you (Recommandé pour vous), la suppression d'un jeu de données interrompt les mises à jour automatiques pour les versions de solution ou les recommandations associées.
- Une ressource `DatasetImportJob` associée ne peut pas avoir l'état CREATE PENDING ou IN PROGRESS.
- Non associé `BatchInferenceJob` ou ne `BatchSegmentJob` peut avoir le statut CREATE PENDING ou EN COURS.
- Non associé`Recommender`, `SolutionVersion` peut avoir le statut CREATE PENDING ou EN COURS.
- Aucun associé ne `Campaign` peut avoir le statut CREATE PENDING, EN COURS ou ACTIF.

Ordre recommandé pour la suppression des ressources

Pour éviter les erreurs de suppression, nous vous recommandons de supprimer les ressources d'un groupe de jeux de données dans l'ordre suivant. Pour identifier les ressources d'un groupe de jeux de données, vous pouvez utiliser les opérations de l'API List. Par exemple, vous pouvez utiliser l'opération ListFilters API pour identifier tous les filtres d'un groupe de jeux de données.

1. Toute campagne ou recommandation : pour supprimer votre campagne ou votre recommandateur à l'aide du APIs, utilisez les opérations de l'DeleteRecommenderAPI DeleteCampaign or. Avec les recommandations, vous pouvez arrêter une recommandation et la démarrer plus tard. Ainsi, vous pouvez suspendre la facturation des recommandations et ne payer que lorsque vous l'utilisez. Pour de plus amples informations, veuillez consulter Arrêter un recommandeur.
2. Toutes les solutions : pour supprimer votre solution avec le APIs, utilisez l'opération DeleteSolution API. Pour supprimer une solution, aucune mise à jour de la solution ne peut être en cours. Son `latestSolutionUpdate` statut doit être ACTIF ou CREATE FAILED. La suppression d'une solution entraîne la suppression de toutes les versions de solution associées. Aucune des versions de sa solution ne peut avoir le statut CREATE PENDING ou EN COURS.
3. Suivi des événements : pour supprimer un outil de suivi des événements avec le APIs, utilisez l'opération DeleteEventTracker API. Vous devez supprimer votre outil de suivi des événements avant de pouvoir supprimer un ensemble de données d'interactions entre objets.
4. Attribution métrique — Pour supprimer une attribution métrique avec le APIs, utilisez l'opération DeleteMetricAttribution API.
5. Tous les filtres — Pour supprimer un filtre avec le APIs, utilisez l'opération DeleteFilter API. Vous devez supprimer tous les filtres avant de supprimer un ensemble de données.
6. Tous les ensembles de données : pour supprimer un ensemble de données avec le APIs, utilisez l'opération DeleteDataset API.
7. Groupe de jeux de données : pour supprimer votre groupe de jeux de données à l'aide du APIs, utilisez l'opération DeleteDatasetGroup API.
8. Schémas — Pour supprimer un schéma, utilisez l'opération DeleteSchema API.

Supprimer des utilisateurs et leurs données à l'aide d'une tâche de suppression de données

Après avoir importé des données, vous pouvez supprimer les utilisateurs et leurs données, y compris leurs métadonnées et leurs données d'interaction, d'un groupe de jeux de données. Vous pouvez supprimer des données utilisateur dans le cadre d'un programme de conformité, ou pour répondre aux demandes de suppression d'utilisateurs, ou pour maintenir vos données à jour en fonction de l'évolution de votre base d'utilisateurs.

Une fois que vous avez supprimé des utilisateurs, Amazon Personalize ne s'entraîne plus sur leurs données et ne prend plus en compte les utilisateurs lors de la génération de segments d'utilisateurs.

Pour supprimer les références aux utilisateurs dans les ensembles de données Amazon Personalize et les modèles d'un groupe de jeux de données, procédez comme suit :

1. Préparez un fichier CSV répertoriant les UserID des utilisateurs à supprimer dans une colonne USER_ID.
2. Téléchargez le fichier CSV dans un compartiment Amazon S3. Votre rôle de service Amazon Personalize doit être autorisé à accéder à ce compartiment.
3. Créez une tâche de suppression de données. Une tâche de suppression de données est une tâche par lots qui supprime les utilisateurs et leurs données des modèles et des ensembles de données d'un groupe de jeux de données.

Rubriques

- Directives et exigences
- Préparation d'une liste d'utilisateurs à supprimer
- Création d'une tâche de suppression de données

Directives et exigences

Les instructions et les exigences relatives à la suppression d'utilisateurs sont les suivantes :

- Avant de créer une tâche de suppression de données, assurez-vous qu'aucune tâche utilisant vos ensembles de données n'est en cours, comme les tâches de formation, les tâches par lots ou les opérations d'importation groupées ou individuelles. Et évitez de créer de tels emplois pendant

qu'une tâche de suppression de données est en cours. En cas de formation ou d'importation, nous ne pouvons pas garantir que les données des utilisateurs seront supprimées des modèles et nous vous recommandons de créer une tâche de suppression de données supplémentaire.

- Une tâche de suppression de données ne supprime pas les références à des utilisateurs extérieurs à Amazon Personalize. Par exemple, il ne supprime pas leur UserID des recommandations par lots dans votre compartiment Amazon S3. Vous devez supprimer ces enregistrements manuellement.
- Vous pouvez avoir jusqu'à 5 tâches de suppression pour un groupe de jeux de données dont le statut est EN ATTENTE.
- La taille totale maximale de votre ou de vos fichiers d'entrée de suppression de données est de 100 Mo. Vous pouvez réutiliser le même fichier d'entrée lorsque vous créez des tâches de suppression.
- Chaque tâche de suppression de données supprime les utilisateurs et leurs données d'interaction dans un groupe de jeux de données. Pour supprimer leurs données dans tous les groupes de jeux de données, vous devez créer une tâche de suppression de données pour chaque groupe de jeux de données.
- Après avoir créé une tâche, la suppression des données des utilisateurs des ensembles de données et des modèles peut prendre jusqu'à un jour.
- Une fois la tâche terminée, veillez à mettre à jour les ressources personnalisées. Assurez-vous de créer une nouvelle version de la solution et, si nécessaire, de mettre à jour votre campagne. Si vous utilisez l'entraînement automatique, vous pouvez toujours créer manuellement de nouvelles versions de solution.
- Votre rôle de service Amazon Personalize doit être autorisé à accéder à votre compartiment Amazon S3 contenant la liste des utilisateurs à supprimer. Il a besoin `GetObject` d'`ListBucket`autorisations et d'autorisations pour le bucket et son contenu. Ces autorisations sont identiques à celles relatives à l'importation de données. Pour plus d'informations sur l'octroi d'autorisations et des exemples de politiques, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.
- Vous ne pouvez pas utiliser votre propre AWS Key Management Service clé dans le compartiment Amazon S3 qui stocke la liste des UserID des utilisateurs à supprimer.
- Si un élément apparaît uniquement dans votre jeu de données d'interactions entre articles et que seuls les utilisateurs que vous supprimez ont interagi avec cet élément, cet élément n'apparaîtra plus dans les recommandations.

Préparation d'une liste d'utilisateurs à supprimer

Avant de supprimer des utilisateurs d'Amazon Personalize, vous devez préparer une liste d'utilisateurs à supprimer dans un fichier CSV et la télécharger sur Amazon S3.

Pour préparer la liste des utilisateurs à supprimer et à télécharger

1. Créez un fichier CSV répertoriant les identifiants des utilisateurs à supprimer. Ce qui suit montre comment votre fichier CSV doit être formaté.

   ```
   USER_ID
   abc
   2a
   5basc
   ab35
   123f
   a55d
   0v22
   441fa
   efg
   ```

2. Chargez votre fichier CSV dans un compartiment Amazon Simple Storage Service (Amazon S3). Pour plus d'informations sur le chargement de fichiers vers Amazon S3, consultez la section Chargement de fichiers et de dossiers par glisser-déposer dans le guide de l'utilisateur d'Amazon Simple Storage Service.
3. Donnez à Amazon Personalize l'accès à votre bucket et à votre fichier CSV. Amazon Personalize doit être autorisé à exécuter les `ListBucket` actions `GetObject` et sur votre compartiment et son contenu. Ces autorisations sont identiques à celles relatives à l'importation de données. Pour plus d'informations sur l'octroi d'autorisations et des exemples de politiques, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Création d'une tâche de suppression de données

Une fois que vous avez terminéPréparation d'une liste d'utilisateurs à supprimer, vous êtes prêt à supprimer les utilisateurs à l'aide d'une tâche de suppression de données.

Une tâche de suppression de données est une tâche par lots qui supprime les utilisateurs et leurs données des modèles et des ensembles de données d'un groupe de jeux de données. Une fois que

vous avez supprimé des utilisateurs, Amazon Personalize ne s'entraîne plus sur leurs données et ne prend plus en compte les utilisateurs lors de la génération de segments d'utilisateurs.

Lorsque vous créez une tâche de suppression de données, vous spécifiez l'emplacement Amazon S3 de votre liste d'utilisateurs à supprimer.

- Si vos données se trouvent dans un seul fichier, utilisez la syntaxe suivante pour l'emplacement Amazon S3 :

  `s3://amzn-s3-demo-bucket/<folder path>/<CSV filename>.csv`
- Si vos fichiers CSV se trouvent dans un dossier de votre compartiment Amazon S3, vous pouvez spécifier le chemin d'accès au dossier. Dans le cadre d'une tâche de suppression de données, Amazon Personalize utilise tous les fichiers portant l'extension de `.csv` fichier dans le dossier et dans n'importe quel sous-dossier. Il ignore les fichiers de tout autre type. Utilisez la syntaxe suivante avec un `/` après le nom du dossier :

  `s3://amzn-s3-demo-bucket/<folder path>/`

Le rôle que vous utilisez doit être autorisé à exécuter les `ListBucket` actions `GetObject` et sur votre compartiment Amazon S3 et son contenu. Pour plus d'informations sur l'octroi d'autorisations et des exemples de politiques, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Vous pouvez créer une tâche de suppression de données à l'aide de la console Amazon Personalize, du AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Création d'une tâche de suppression de données (console)

Pour supprimer des utilisateurs à l'aide de la console Amazon Personalize, créez une tâche de suppression de données avec un nom, le rôle de service IAM et l'emplacement Amazon S3 de vos données.

Pour supprimer des enregistrements (console)

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données. La vue d'ensemble du groupe de jeux de données s'affiche.
3. Dans le volet de navigation, sélectionnez Datasets.

4. Dans Tâches de suppression de données, choisissez Créer une tâche.
5. Dans Détails du job, nommez le job.
6. Dans la source d'entrée S3, pour S3 Location, spécifiez l'emplacement Amazon S3 du fichier CSV qui stocke la liste des identifiants utilisateur des utilisateurs à supprimer. Vous avez préparé ce fichier enPréparation d'une liste d'utilisateurs à supprimer.
7. Dans le rôle IAM, choisissez de créer un nouveau rôle ou d'utiliser un rôle existant. Si vous avez rempli les conditions requises pour créer un rôle pour Amazon Personalize et que vous avez accordé à ce rôle l'accès à votre compartiment Amazon S3, choisissez Utiliser un rôle de service existant et spécifiez le rôle dans Création d'un rôle IAM pour Amazon Personalize lequel vous l'avez créé.

   Le rôle que vous utilisez doit être autorisé à exécuter les `ListBucket` actions `GetObject` et sur votre compartiment Amazon S3 et son contenu. Ces autorisations sont identiques à celles relatives à l'importation de données. Pour plus d'informations sur l'octroi d'autorisations et des exemples de politiques, consultezPermettre à Amazon Personalize d'accéder aux ressources Amazon S3.
8. Pour les balises, ajoutez éventuellement des balises. Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.
9. Choisissez Créer une tâche. La tâche démarre et la page de détails s'affiche.

   Après avoir créé une tâche, il faut environ une journée pour supprimer les données des utilisateurs des ensembles de données et des modèles. Jusqu'à ce que le travail soit terminé, Amazon Personalize continue d'utiliser les données lors de la formation. Et les utilisateurs peuvent apparaître dans des segments d'utilisateurs.

   La suppression des données est terminée lorsque le statut indique TERMINÉ. Si la tâche échoue pour une raison quelconque, nous vous recommandons de créer une autre tâche de suppression de données. Une fois la tâche terminée, veillez à mettre à jour les ressources personnalisées. Assurez-vous de créer une nouvelle version de la solution et, si nécessaire, de mettre à jour votre campagne. Si vous utilisez l'entraînement automatique, vous pouvez toujours créer manuellement de nouvelles versions de solution.

Création d'une tâche de suppression de données (AWS CLI)

Pour supprimer des utilisateurs avec le AWS CLI, utilisez la `create-data-deletion-job` commande. Cette commande utilise l'opération CreateDataDeletion API. Le code suivant montre comment créer une tâche de suppression de données. Pour utiliser le code, mettez-le à jour pour

spécifier le nom de la tâche, le rôle IAM dans Création d'un rôle IAM pour Amazon Personalize lequel vous l'avez créé et l'emplacement de vos données sur Amazon S3. Vous avez préparé ce fichier enPréparation d'une liste d'utilisateurs à supprimer.

```
aws personalize create-data-deletion-job \
--job-name deletion job name \
--dataset-group-arn dataset group ARN \
--data-source dataLocation=s3://amzn-s3-demo-bucket/filename.csv \
--role-arn roleArn
```

Après avoir créé une tâche, il faut environ une journée pour supprimer les données des utilisateurs des ensembles de données et des modèles. Jusqu'à ce que le travail soit terminé, Amazon Personalize continue d'utiliser les données lors de la formation. Et les utilisateurs peuvent apparaître dans des segments d'utilisateurs.

La tâche est terminée lorsque le statut est TERMINÉ. Vérifiez l'état à l'aide de la `describe-data-deletion-job` commande et spécifiez l'ARN de la tâche de suppression des données. Pour plus d'informations sur le fonctionnement de l'API, consultezDescribeDataDeletionJob. Pour consulter l'historique des tâches de suppression de données triées par date de création, utilisez l'opération ListDataDeletionJobs API.

Si la tâche échoue pour une raison quelconque, nous vous recommandons de créer une autre tâche de suppression de données. Une fois la tâche terminée, veillez à mettre à jour les ressources personnalisées. Assurez-vous de créer une nouvelle version de la solution et, si nécessaire, de mettre à jour votre campagne. Si vous utilisez l'entraînement automatique, vous pouvez toujours créer manuellement de nouvelles versions de solution.

Création d'une tâche de suppression de données (AWS SDKs)

Pour supprimer des utilisateurs avec le AWS SDKs, utilisez l'opération CreateDataDeletionJob API. Le code suivant montre comment créer une tâche de suppression de données. Pour utiliser le code, mettez-le à jour pour spécifier le nom de la tâche, le rôle IAM dans Création d'un rôle IAM pour Amazon Personalize lequel vous l'avez créé et l'emplacement de vos données sur Amazon S3. Vous avez préparé ce fichier enPréparation d'une liste d'utilisateurs à supprimer.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_data_deletion_job(
```

```
    jobName = 'Deletion job name',
    datasetGroupArn = 'Dataset Group ARN',
    dataSource = {'dataLocation':'s3://amzn-s3-demo-bucket/file.csv'},
    roleArn = 'role_arn'
)

deletion_job_arn = response['dataDeletionJobArn']

print ('Deletion Job arn: ' + deletion_job_arn)

description = personalize.describe_data_deletion_job(
    dataDeletionJobArn = deletion_job_arn)['dataDeletionJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['dataDeletionJobArn'])
print('Status: ' + description['status'])
```

Après avoir créé une tâche, il faut environ une journée pour supprimer les données des utilisateurs des ensembles de données et des modèles. Jusqu'à ce que le travail soit terminé, Amazon Personalize continue d'utiliser les données lors de la formation. Et les utilisateurs peuvent apparaître dans des segments d'utilisateurs.

La tâche est terminée lorsque le statut est TERMINÉ. Vérifiez l'état à l'aide de l'DescribeDataDeletionJobopération et spécifiez l'ARN de la tâche de suppression des données. Pour consulter l'historique des tâches de suppression de données triées par date de création, utilisez l'opération ListDataDeletionJobs API.

Si la tâche échoue pour une raison quelconque, nous vous recommandons de créer une autre tâche de suppression de données. Une fois la tâche terminée, veillez à mettre à jour les ressources personnalisées. Assurez-vous de créer une nouvelle version de la solution et, si nécessaire, de mettre à jour votre campagne. Si vous utilisez l'entraînement automatique, vous pouvez toujours créer manuellement de nouvelles versions de solution.

Supprimer un ensemble de données pour supprimer toutes ses données

Pour supprimer toutes les données d'un ensemble de données, vous devez supprimer le jeu de données. Vous pouvez supprimer un ensemble de données à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Avant de supprimer un ensemble de données, tenez compte des points suivants :

- Vous devez supprimer tous les filtres avant de supprimer un ensemble de données.
- Si vous avez créé un outil de suivi des événements, vous devez le supprimer avant de supprimer un ensemble de données d'interactions entre les éléments.
- Si vous avez créé une attribution métrique qui fait référence au jeu de données, vous devez d'abord supprimer l'attribution métrique.
- Si vous utilisez User-Personalization-v 2 scénarios d'utilisation (Personnalisation utilisateur) ou Next-Best-Action recettes ou Top picks for you et Recommended for you (Recommandé pour vous), la suppression d'un jeu de données interrompt les mises à jour automatiques pour les versions de solution ou les recommandations associées.
- Une ressource `DatasetImportJob` associée ne peut pas avoir l'état CREATE PENDING ou IN PROGRESS.
- Non associé `BatchInferenceJob` ou ne `BatchSegmentJob` peut avoir le statut CREATE PENDING ou EN COURS.
- Non associé`Recommender`, `SolutionVersion` peut avoir le statut CREATE PENDING ou EN COURS.
- Aucun associé ne `Campaign` peut avoir le statut CREATE PENDING, EN COURS ou ACTIF.

Rubriques

- Supprimer un ensemble de données (console)
- Supprimer un ensemble de données (AWS CLI)
- Supprimer un ensemble de données (AWS SDKs)

Supprimer un ensemble de données (console)

Pour supprimer un ensemble de données à l'aide de la console Amazon Personalize, accédez à la page de détails du jeu de données et choisissez Supprimer.

Supprimer un jeu de données

1. Ouvrez la console Amazon Personalize à la https://console.aws.amazon.com/personalize/maison.
2. Dans le volet de navigation, sélectionnez Groupes de jeux de données.
3. Sur la page Groupes de jeux de données, choisissez votre groupe de jeux de données.

4. Dans le volet de navigation, sélectionnez Datasets.
5. Choisissez le jeu de données pour ouvrir sa page de détails.
6. Sur la page de détails de l'ensemble de données, choisissez Supprimer et confirmez la suppression de l'ensemble de données.

Supprimer un ensemble de données (AWS CLI)

Le code suivant montre comment supprimer un ensemble de données à l'aide de l'DeleteDatasetopération AWS CLI et.

```
aws personalize delete-dataset --dataset-arn dataset-arn
```

Supprimer un ensemble de données (AWS SDKs)

Le code suivant montre comment supprimer un ensemble de données à l'aide de l'DeleteDatasetopération AWS SDKs et.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.delete_dataset(
    datasetArn = 'dataset ARN'
)
```

SDK for Java 2.x

```
public static void deleteDataset(PersonalizeClient personalizeClient,
                                 String datasetArn) {

    try {
        DeleteDatasetRequest deleteRequest = DeleteDatasetRequest.builder()
                .datasetArn(datasetArn)
                .build();

        int responseCode =
 personalizeClient.deleteDataset(deleteRequest).sdkHttpResponse().statusCode();
```

```
        System.out.println(responseCode);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Mesurer l'impact des recommandations d'Amazon Personalize

Au fur et à mesure que vos clients interagissent avec les recommandations, vous pouvez mesurer dans quelle mesure les recommandations d'Amazon Personalize vous aident à atteindre vos objectifs. Vous pouvez identifier les campagnes et les recommandeurs qui ont le plus d'impact sur les indicateurs de performance clés. Par exemple, vous pouvez identifier la ressource qui génère le plus de minutes regardées, le plus de clics ou le plus d'achats. Vous pouvez également comparer les performances des recommandations d'Amazon Personalize à celles générées par des services tiers.

Lorsque vous savez quelle campagne ou quel agent de recommandation génère le plus d'impact, vous pouvez prendre des mesures pour tirer davantage parti de ses recommandations. Par exemple, vous pouvez augmenter la visibilité des recommandations sur votre site afin de susciter davantage d'engagement. Vous pouvez également intégrer les recommandations dans une campagne marketing, telle que des e-mails personnalisés ou des publicités ciblées.

Si vous identifiez une ressource qui n'a pas l'impact escompté, vous pouvez prendre des mesures pour améliorer les recommandations. Par exemple, vous pouvez utiliser la console Amazon Personalize pour analyser les données d'entraînement utilisées pour créer la ressource, apporter les améliorations de données recommandées, puis réimporter les données. Pour plus d'informations sur l'analyse des données, consultezAnalyse de la qualité et de la quantité de données dans les ensembles de données Amazon Personalize.

Les éléments suivants peuvent vous aider à mesurer l'impact des recommandations :

- Attribution métrique : une attribution métrique Amazon Personalize crée des rapports basés sur les métriques que vous spécifiez, les interactions entre les articles et les données sur les articles que vous importez. Par exemple, la durée totale des films visionnés par les utilisateurs ou le nombre total d'événements liés aux clics. Une fois que vous avez créé une attribution de métriques, Amazon Personalize envoie automatiquement des métriques sur les événements liés à l'opération d'PutEventsAPI et des données en masse incrémentielles à Amazon CloudWatch. Pour les données en masse, vous pouvez choisir de publier des rapports dans un compartiment Amazon S3.
- Tests A/B : effectuer des A/B tests avec les recommandations d'Amazon Personalize implique de présenter différents types de recommandations à différents groupes d'utilisateurs et de comparer les résultats. Vous pouvez utiliser A/B les tests pour comparer et évaluer différentes stratégies de recommandation, évaluer les performances du modèle et mesurer l'impact des recommandations.

Rubriques

- Mesurer l'impact des recommandations à l'aide d'une attribution métrique
- Mesurer l'impact des recommandations à l'aide A/B de tests

Mesurer l'impact des recommandations à l'aide d'une attribution métrique

Pour mesurer l'impact des recommandations relatives aux articles, vous pouvez créer une attribution métrique. Une attribution de mesures crée des rapports basés sur les interactions entre les articles et les données sur les articles que vous importez, ainsi que sur les mesures que vous spécifiez. Par exemple, la durée totale des films visionnés par les utilisateurs ou le nombre total d'événements liés aux clics. Amazon Personalize agrège les calculs sur une fenêtre de 15 minutes. Pour les données d'interaction diffusées en continu et les données en masse incrémentielles, Amazon Personalize envoie automatiquement des rapports métriques à Amazon. CloudWatch Pour les données en masse, vous pouvez choisir de publier des rapports dans un compartiment Amazon S3.

Pour chaque interaction que vous importez, incluez des données source afin de comparer différentes campagnes, recommandations et tiers. Vous pouvez inclure l'ID de recommandation des recommandations que vous avez présentées à l'utilisateur ou à la source de l'événement, telle qu'un tiers.

Par exemple, vous avez peut-être une application de streaming vidéo qui affiche des recommandations de films provenant de deux recommandations Amazon Personalize différents. Si vous souhaitez savoir quel recommandeur génère le plus grand nombre d'événements surveillés, vous pouvez créer une attribution métrique qui permet de suivre le nombre total d'événements surveillés. Vous pouvez ensuite enregistrer les événements de surveillance au fur et à mesure que les utilisateurs interagissent avec les recommandations, et les inclure `recommendationId` dans chaque événement. Amazon Personalize utilise le `recommendationId` pour identifier chaque recommandeur. Lorsque vous enregistrez des événements, vous pouvez consulter le total des événements regardés agrégés toutes les 15 minutes pour les deux recommandateurs présents. CloudWatch Pour des exemples de code montrant comment inclure a `recommendationId` ou an `eventAttributionSource` pour un événement, consultezStatistiques relatives aux événements et rapports d'attribution.

Rubriques

- Directives et exigences relatives à l'attribution d'une métrique

- Création d'une attribution métrique Amazon Personalize
- Mettre à jour l'attribution d'une métrique Amazon Personalize
- Supprimer une attribution de métrique Amazon Personalize
- Affichage de graphiques de données métriques dans CloudWatch
- Publication de rapports d'attribution de métriques sur Amazon S3

Directives et exigences relatives à l'attribution d'une métrique

Amazon Personalize ne commence à calculer et à signaler l'impact des recommandations qu'une fois que vous avez créé une attribution métrique. Pour créer l'historique le plus complet possible, nous vous recommandons de créer une attribution métrique avant d'importer vos données d'interactions. Lorsque vous créez une tâche d'importation d'un ensemble de données pour un ensemble de données d'interactions entre articles à l'aide de la console Amazon Personalize, vous avez la possibilité de créer une attribution de métrique dans un nouvel onglet. Vous pouvez ensuite retourner à la tâche d'importation pour la terminer.

Une fois que vous avez créé une attribution de métrique et enregistré des événements ou importé des données en masse incrémentielles, vous devrez payer un certain CloudWatch coût mensuel par métrique. Pour plus d'informations sur CloudWatch les tarifs, consultez la page de CloudWatch tarification d'Amazon. Pour arrêter d'envoyer des métriques à CloudWatch, supprimez l'attribution des métriques.

Pour évaluer l'impact des recommandations au fil du temps, continuez à importer des données au fur et à mesure que les clients interagissent avec les recommandations. Si vous avez déjà importé des données, vous pouvez toujours créer une attribution métrique et commencer à mesurer l'impact des recommandations. Toutefois, Amazon Personalize ne signalera pas les données que vous avez importées avant de les créer.

Les directives et les exigences relatives à la génération de rapports avec une attribution métrique sont les suivantes :

- Vous devez autoriser Amazon Personalize à accéder aux données et à les saisir CloudWatch. Pour obtenir des exemples de stratégie, consultez Donner à Amazon Personalize l'accès à CloudWatch.
- Pour publier des métriques sur Amazon S3, autorisez Amazon Personalize à écrire dans votre compartiment. Vous devez également indiquer le chemin du compartiment dans votre attribution de métriques. Pour obtenir des exemples de stratégie, consultez Permettre à Amazon Personalize d'accéder à votre compartiment Amazon S3.

- Pour publier des statistiques sur CloudWatch, les enregistrements doivent dater de moins de 14 jours. Si vos données sont anciennes, ces enregistrements ne seront pas inclus dans les calculs ou les rapports.
- L'importation d'événements dupliqués (événements correspondant exactement à tous les attributs) peut entraîner des comportements inattendus, notamment des indicateurs inexacts. Nous vous recommandons de supprimer les enregistrements dupliqués de toutes les données en bloc avant l'importation et d'éviter d'importer des événements dupliqués lors de l'`PutEvents`opération.
- Votre jeu de données sur les interactions entre objets doit comporter une `EVENT_TYPE` colonne.
- Vous ne pouvez pas créer de rapports métriques pour les données d'un ensemble de données d'interactions Action.
- Vous pouvez créer au maximum une attribution de mesures par groupe de jeux de données. Chaque attribution de métrique peut comporter au maximum 10 métriques.

Pour comparer les sources, chaque événement d'interaction doit inclure un `recommendationId` ou`eventAttributionSource`. Vous pouvez fournir au maximum 100 sources d'attribution d'événements uniques. Pour obtenir des exemples de code `PutEvents`, consultez Statistiques relatives aux événements et rapports d'attribution.

- Si vous fournissez un`recommendationId`, Amazon Personalize détermine automatiquement la campagne ou le recommandateur source et l'identifie dans les rapports d'une colonne EVENT_ATTRIBUTION_SOURCE.
- Si vous fournissez les deux attributs, Amazon Personalize utilise uniquement le`eventAttributionSource`.
- Si vous ne fournissez pas de source, Amazon Personalize l'étiquette `SOURCE_NAME_UNDEFINED` dans les rapports.

Rubriques

- Donner à Amazon Personalize l'accès à CloudWatch
- Permettre à Amazon Personalize d'accéder à votre compartiment Amazon S3

Donner à Amazon Personalize l'accès à CloudWatch

> **Important**
>
> Lorsque vous accordez des autorisations, Amazon Personalize place et valide une petite quantité de données. CloudWatch Cela entraînera un coût unique de moins de 0,30$. Pour plus d'informations sur CloudWatch les tarifs, consultez la page de CloudWatch tarification d'Amazon.

Pour donner accès à Amazon Personalize CloudWatch, associez une nouvelle politique Gestion des identités et des accès AWS (IAM) à votre rôle de service Amazon Personalize qui accorde au rôle l'autorisation d'utiliser l'`PutMetricData`action pour CloudWatch. L'exemple de politique suivant octroie `PutMetricData` des autorisations.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    }
  ]
}
```

Permettre à Amazon Personalize d'accéder à votre compartiment Amazon S3

Pour autoriser Amazon Personalize à accéder à votre compartiment Amazon S3, procédez comme suit :

- Associez une politique IAM à votre rôle de service Amazon Personalize qui accorde au rôle l'autorisation d'utiliser l'`PutObject`action sur votre compartiment.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

- Attachez une politique de compartiment à votre compartiment Amazon S3 de sortie qui accorde au principe Amazon Personalize l'autorisation d'utiliser les `PutObject` actions.

  Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

  JSON

```
{
    "Version":"2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Création d'une attribution métrique Amazon Personalize

> **Important**
>
> Une fois que vous avez créé une attribution de métrique et enregistré des événements ou importé des données en masse incrémentielles, vous devrez payer un certain CloudWatch coût mensuel par métrique. Pour plus d'informations sur CloudWatch les tarifs, consultez la page de CloudWatch tarification d'Amazon. Pour arrêter d'envoyer des métriques à CloudWatch, supprimez l'attribution des métriques.

Pour commencer à générer des rapports de mesures, vous devez créer une attribution de métrique et importer des données d'interactions. Lorsque vous créez une attribution de métrique, vous spécifiez une liste de types d'événements à signaler. Pour chaque type d'événement, vous spécifiez une fonction qu'Amazon Personalize applique lors de la collecte des données. Les fonctions disponibles incluent `SUM(DatasetType.COLUMN_NAME)` et`SAMPLECOUNT()`.

Par exemple, vous avez peut-être une application de streaming vidéo en ligne et souhaitez suivre deux indicateurs : le taux de clics pour les recommandations et la durée totale des films visionnés, chaque vidéo du jeu de données Items incluant un `LENGTH` attribut. Vous créeriez une attribution de métriques et ajouteriez deux métriques, chacune associée à un type d'événement et à une fonction. Le premier peut concerner le type d'`Click`événement avec une `SAMPLECOUNT()` fonction. La seconde peut concerner le type d'`Watch`événement avec une `SUM(Items.LENGTH)` fonction.

Vous pouvez appliquer `SUM()` des fonctions uniquement aux colonnes numériques des éléments et aux ensembles de données d'interactions entre éléments. Pour appliquer une `SUM()` fonction à une colonne d'un jeu de données Items, vous devez d'abord importer les métadonnées des éléments.

Vous pouvez créer une attribution de métriques à l'aide de la console Amazon Personalize ou AWS des SDK. AWS Command Line Interface

Rubriques

- Création d'une attribution métrique (console)
- Création d'une attribution métrique (AWS CLI)
- Création d'une attribution métrique (AWS SDK)

Création d'une attribution métrique (console)

Pour créer une attribution métrique avec la console Amazon Personalize, accédez à la page d'attribution métrique et choisissez Create metric attribution. Lorsque vous créez une attribution de métrique, vous spécifiez un chemin de compartiment Amazon S3 facultatif, votre rôle de service Amazon Personalize IAM et une liste de métriques sur lesquelles établir des rapports.

Lorsque vous créez une tâche d'importation d'un ensemble de données Item interactions avec la console Amazon Personalize, vous avez la possibilité de créer une attribution métrique dans un nouvel onglet. Vous pouvez ensuite retourner à la tâche d'importation pour la terminer. Si vous êtes déjà sur la page Configurer l'attribution des métriques, vous pouvez passer à l'étape 4.

Pour créer une attribution métrique

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sous Ressources personnalisées, sélectionnez Attribution métrique.
4. Dans Détails de l'attribution métrique, choisissez Créer une attribution métrique.
5. Sur la page Configurer l'attribution des métriques, nommez l'attribution des métriques.
6. Si vous souhaitez publier des métriques sur Amazon S3 pour le chemin de sortie des données Amazon S3, entrez le compartiment Amazon S3 de destination. Cela permet de publier des métriques chaque fois que vous créez une tâche d'importation de jeu de données. Utilisez la syntaxe suivante :

   ```
   s3://amzn-s3-demo-bucket/<folder> path>
   ```

7. Si vous utilisez AWS KMS pour le chiffrement, pour l'ARN de la clé KMS, entrez le nom de ressource Amazon (ARN) de la AWS KMS clé. Vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d’informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

8. Dans le rôle IAM, choisissez de créer un nouveau rôle de service ou d'utiliser un rôle existant. Le rôle que vous choisissez doit disposer `PutMetricData` d'autorisations pour CloudWatch. Si vous souhaitez publier sur Amazon S3, le rôle doit disposer d'`PutObject`autorisations pour votre compartiment Amazon S3.

   Pour utiliser le rôle que vous avez créé dansCréation d'un rôle IAM pour Amazon Personalize, vous devrez peut-être ajouter des politiques pour CloudWatch Amazon S3.

   Pour des exemples de politiques, voir Donner à Amazon Personalize l'accès à CloudWatch etPermettre à Amazon Personalize d'accéder à votre compartiment Amazon S3.

9. Choisissez Suivant.

10. Sur la page Définir les attributs des métriques, choisissez comment définir les métriques. Choisissez Créer des attributs métriques pour utiliser l'outil de création. Choisissez Input metric attributes pour saisir les métriques au format JSON.

    - Si vous choisissez Créer des attributs de métrique, pour chaque métrique, indiquez un nom, un type d'événement et choisissez une fonction. Pour les `SUM( )` fonctions, choisissez le nom de la colonne. Choisissez Ajouter un attribut de métrique pour ajouter des métriques supplémentaires.
    - Si vous choisissez Attributs de métrique en entrée, entrez chaque métrique au format JSON. Ce qui suit montre comment formater une métrique.

      ```
      {
          "EventType": "watch",
          "MetricName": "MinutesWatchedTracker",
          "MetricMathExpression": "SUM(Items.LENGTH)"
      }
      ```

11. Choisissez Suivant.

12. Sur la page Réviser et créer, passez en revue les détails de la nouvelle attribution de métriques. Pour effectuer des modifications, sélectionnez Précédent. Pour créer l'attribution métrique, choisissez Create. Lorsque l'attribution des métriques est active, vous pouvez commencer à importer des données et consulter les résultats. Pour plus d'informations sur l'affichage des résultats, consultezAffichage de graphiques de données métriques dans CloudWatch. Pour plus d'informations sur la publication des résultats sur Amazon S3, consultezPublication de rapports d'attribution de métriques sur Amazon S3.

Création d'une attribution métrique (AWS CLI)

Le code suivant montre comment créer une attribution métrique avec le AWS Command Line Interface. Le rôle que vous spécifiez doit disposer `PutMetricData` d'autorisations CloudWatch et, en cas de publication sur Amazon S3, `PutObject` d'autorisations pour votre compartiment Amazon S3. Pour utiliser le rôle que vous avez créé dansCréation d'un rôle IAM pour Amazon Personalize, vous devrez peut-être ajouter des politiques pour CloudWatch Amazon S3. Pour des exemples de politiques, voir Donner à Amazon Personalize l'accès à CloudWatch etPermettre à Amazon Personalize d'accéder à votre compartiment Amazon S3.

Pour chaque métrique, spécifiez un nom, un type d'événement et une expression (une fonction). Les fonctions disponibles incluent `SUM(DatasetType.COLUMN_NAME)` et`SAMPLECOUNT()`. Pour les fonctions SUM (), spécifiez le type de jeu de données et le nom de colonne. Par exemple : `SUM(Items.LENGTH)`. Pour plus d'informations sur chaque paramètre, consultezCreateMetricAttribution.

```
aws personalize create-metric-attribution \
--name metric attribution name \
--dataset-group-arn dataset group arn \
--metrics-output-config "{\"roleArn\": \"Amazon Personalize service role ARN\",
 \"s3DataDestination\":{\"kmsKeyArn\":\"kms key ARN\",\"path\":\"s3://amzn-s3-demo-
bucket/folder-name/\"}}" \
--metrics "[{
  \"eventType\": \"event type\",
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",
  \"metricName\": \"metric name\"
}]"
```

Création d'une attribution métrique (AWS SDK)

Le code suivant montre comment créer une attribution métrique avec le AWS SDK pour Python (Boto3). Le rôle que vous spécifiez doit disposer `PutMetricData` d'autorisations CloudWatch et, en cas de publication sur Amazon S3, `PutObject` d'autorisations pour votre compartiment Amazon S3. Pour utiliser le rôle que vous avez créé dansCréation d'un rôle IAM pour Amazon Personalize, vous devrez peut-être ajouter des politiques pour CloudWatch Amazon S3. Pour des exemples de politiques, voir Donner à Amazon Personalize l'accès à CloudWatch etPermettre à Amazon Personalize d'accéder à votre compartiment Amazon S3.

Pour chaque métrique, spécifiez un nom, un type d'événement et une expression (une fonction). Les fonctions disponibles incluent `SUM(DatasetType.COLUMN_NAME)` et`SAMPLECOUNT()`.

Pour les fonctions SUM (), spécifiez le type de jeu de données et le nom de colonne. Par exemple : `SUM(Items.LENGTH)`. Pour plus d'informations sur chaque paramètre, consultezCreateMetricAttribution.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsList = [{
      "eventType": "event type",
      "expression": "SUM(DatasetType.COLUMN_NAME)",
      "metricName": "metric name"
}]

outputConfig = {
  "roleArn": "Amazon Personalize service role ARN",
  "s3DataDestination": {
    "kmsKeyArn": "key ARN",
    "path": "s3://amzn-s3-demo-bucket/<folder>"
  }
}
response = personalize.create_metric_attribution(
  name = 'metric attribution name',
  datasetGroupArn = 'dataset group arn',
  metricsOutputConfig = outputConfig,
  metrics = metricsList
)

metric_attribution_arn = response['metricAttributionArn']

print ('Metric attribution ARN: ' + metric_attribution_arn)

description = personalize.describe_metric_attribution(
    metricAttributionArn = metric_attribution_arn)['metricAttribution']

print('Name: ' + description['name'])
print('ARN: ' + description['metricAttributionArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createMetricAttribution(PersonalizeClient personalizeClient,
                                             String eventType,
                                             String expression,
                                             String metricName,
                                             String metricAttributionName,
                                             String roleArn,
                                             String s3Path,
                                             String kmsKeyArn,
                                             String datasetGroupArn) {
    String metricAttributionArn = "";

    try {

        MetricAttribute attribute = MetricAttribute.builder()
                .eventType(eventType)
                .expression(expression)
                .metricName(metricName)
                .build();

        ArrayList<MetricAttribute> metricAttributes = new ArrayList<>();
        metricAttributes.add(attribute);

        S3DataConfig s3DataDestination = S3DataConfig.builder()
                .kmsKeyArn(kmsKeyArn)
                .path(s3Path)
                .build();

        MetricAttributionOutput outputConfig = MetricAttributionOutput.builder()
                .roleArn(roleArn)
                .s3DataDestination(s3DataDestination)
                .build();

        CreateMetricAttributionRequest createMetricAttributionRequest =
 CreateMetricAttributionRequest.builder()
                .name(metricAttributionName)
                .datasetGroupArn(datasetGroupArn)
                .metrics(metricAttributes)
                .metricsOutputConfig(outputConfig)
                .build();
        CreateMetricAttributionResponse createMetricAttributionResponse =
 personalizeClient.createMetricAttribution(createMetricAttributionRequest);
```

```
        metricAttributionArn =
 createMetricAttributionResponse.metricAttributionArn();
        System.out.println("Metric attribution ARN: " + metricAttributionArn);
        return metricAttributionArn;
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateMetricAttributionCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the metric attribution param
export const createMetricAttributionParam = {
  name: "METRIC_ATTRIBUTION_NAME",            /* required */
  datasetGroupArn: "DATASET_GROUP_ARN",       /* required */
  metricsOutputConfig: {
    roleArn: "ROLE_ARN",                      /* required */
    s3DataDestination: {
      kmsKeyArn: "KEY_ARN",                                                  /*
 optional */
      path: "s3://amzn-s3-demo-bucket/<folderName>/",    /* optional */
    },
  },
  metrics: [
    {
      eventType: "EVENT_TYPE",                      /* required for each metric */
      expression: "SUM(DatasetType.COLUMN_NAME)",   /* required for each metric */
      metricName: "METRIC_NAME",                    /* required for each metric */
    }
  ]
};

export const run = async () => {
```

```
  try {
    const response = await personalizeClient.send(
      new CreateMetricAttributionCommand(createMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Mettre à jour l'attribution d'une métrique Amazon Personalize

Lorsque vous mettez à jour l'attribution d'une métrique, vous pouvez ajouter et supprimer des métriques et modifier sa configuration de sortie. Vous pouvez mettre à jour l'attribution d'une métrique à l'aide de la console Amazon Personalize ou AWS des SDK. AWS Command Line Interface

Rubriques

- Mettre à jour l'attribution d'une métrique (console)
- Mettre à jour une attribution métrique (AWS CLI)
- Mettre à jour une attribution métrique (AWS SDK)

Mettre à jour l'attribution d'une métrique (console)

Pour mettre à jour l'attribution d'une métrique avec la console Amazon Personalize, vous devez effectuer vos modifications sur la page d'attribution des métriques.

Pour mettre à jour l'attribution d'une métrique

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sélectionnez Attribution métrique.
4. Dans la section inférieure, choisissez l'onglet Attributs métriques ou l'onglet Configuration de l'attribution métrique pour commencer à apporter des modifications.

- Pour ajouter ou supprimer des mesures, cliquez sur l'onglet Attributs des mesures, puis sur Modifier les attributs. Apportez vos modifications sur la page Modifier les attributs des mesures et choisissez Mettre à jour pour enregistrer vos modifications.
- Pour apporter des modifications au compartiment de sortie Amazon S3 ou au rôle de service IAM, choisissez l'onglet Modifier la configuration de l'attribution des métriques et apportez les modifications sur la page Modifier la configuration de l'attribution. Choisissez Mettre à jour pour enregistrer vos modifications.

Mettre à jour une attribution métrique (AWS CLI)

Après avoir créé une attribution de métrique, vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour ajouter et supprimer des métriques et modifier sa configuration de sortie. Le code suivant montre comment supprimer des métriques à l'aide de la `update-metric-attribution` commande :

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--remove-metrics metricName1 metricName2
```

Le code suivant montre comment ajouter une métrique supplémentaire et spécifier une nouvelle configuration de sortie :

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--metrics-output-config "{\"roleArn\": \"new role ARN\", \"s3DataDestination\":
{\"kmsKeyArn\":\"kms key ARN\",\"path\":\"s3://amzn-s3-demo-bucket2/new-folder-name/
\"}}" \
--add-metrics "[{
  \"eventType\": \"event type\",
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",
  \"metricName\": \"metric name\"
}]"
```

En cas de succès, Amazon Personalize renvoie l'ARN de l'attribution métrique que vous avez mise à jour. Pour une liste complète de tous les paramètres, voirUpdateMetricAttribution.

Mettre à jour une attribution métrique (AWS SDK)

Après avoir créé une attribution de métrique, vous pouvez ajouter ou supprimer des métriques et modifier sa configuration de sortie. Le code suivant montre comment supprimer des métriques d'une attribution de métrique.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsToRemove = ["metricName1", "metricName2"]

response = personalize.update_metric_attribution(
  metricAttributionArn = "metric attribution ARN",
  removeMetrics = metricsToRemove
)
```

SDK for Java 2.x

```
public static void removeMetrics(PersonalizeClient client,
                                 String metricAttributionArn,
                                 String metric1Name,
                                 String metric2Name) {

    ArrayList<String> metricsToRemove = new ArrayList<>(Arrays.asList(metric1Name,
 metric2Name));

    try {

        UpdateMetricAttributionRequest request =
 UpdateMetricAttributionRequest.builder()
                .metricAttributionArn(metricAttributionArn)
                .removeMetrics(metricsToRemove)
                .build();

        UpdateMetricAttributionResponse response =
 client.updateMetricAttribution(request);
        System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
```

```
    }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the update request param
export const updateMetricAttributionParam = {
  metricAttributionArn: "METRIC_ATTRIBUTION_ARN",    /* required */
  removeMetrics: ["METRIC_NAME_1", "METRIC_NAME_2"]    /* specify list of names of
 metrics to delete */
};
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new UpdateMetricAttributionCommand(updateMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Le code suivant montre comment ajouter une métrique supplémentaire et spécifier une nouvelle configuration de sortie :

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')
```

```
newMetrics = [{
      "eventType": "event type",
      "expression": "SUM(DatasetType.COLUMN_NAME)",
      "metricName": "metric name"
}]

newOutputConfig = {
  "roleArn": "Amazon Personalize service role ARN",
  "s3DataDestination": {
    "kmsKeyArn": "key ARN",
    "path": "s3://amzn-s3-demo-bucket/<folder>"
  }
}

response = personalize.update_metric_attribution(
  metricAttributionArn = "metric attribution arn",
  metricsOutputConfig = newOutputConfig,
  addMetrics = newMetrics
)
```

SDK for Java 2.x

```
public static void addMetricsAndUpdateOutputConfig(PersonalizeClient
 personalizeClient,
                                                    String metricAttributionArn,
                                                    String newMetric1EventType,
                                                    String newMetric1Expression,
                                                    String newMetric1Name,
                                                    String newMetric2EventType,
                                                    String newMetric2Expression,
                                                    String newMetric2Name,
                                                    String roleArn,
                                                    String s3Path,
                                                    String kmsKeyArn) {
    try {

        MetricAttribute newAttribute = MetricAttribute.builder()
                .eventType(newMetric1EventType)
                .expression(newMetric1Expression)
                .metricName(newMetric1Name)
                .build();

        MetricAttribute newAttribute2 = MetricAttribute.builder()
```

```
                .eventType(newMetric2EventType)
                .expression(newMetric2Expression)
                .metricName(newMetric2Name)
                .build();

        ArrayList<MetricAttribute> newAttributes = new
 ArrayList<>(Arrays.asList(newAttribute, newAttribute2));

        S3DataConfig newDataDestination = S3DataConfig.builder()
                .kmsKeyArn(kmsKeyArn)
                .path(s3Path)
                .build();

        MetricAttributionOutput newOutputConfig = MetricAttributionOutput.builder()
                .roleArn(roleArn)
                .s3DataDestination(newDataDestination)
                .build();

        UpdateMetricAttributionRequest request =
 UpdateMetricAttributionRequest.builder()
                .metricAttributionArn(metricAttributionArn)
                .metricsOutputConfig(newOutputConfig)
                .addMetrics(newAttributes)
                .build();

        UpdateMetricAttributionResponse response =
 personalizeClient.updateMetricAttribution(request);
        System.out.println("New metrics added!");
        System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
```

```
  region: "REGION"
});

export const updateMetricAttributionParam = {
  metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
  addMetrics: [
    {
      eventType: "EVENT_TYPE",                      /* required for each metric */
      expression: "SUM(DatasetType.COLUMN_NAME)",   /* required for each metric */
      metricName: "METRIC_NAME",                    /* required for each metric */
    }
  ],
  metricsOutputConfig: {
    roleArn: "ROLE_ARN",                      /* required */
    s3DataDestination: {
      kmsKeyArn: "KEY_ARN",                                                    /*
 optional */
      path: "s3://amzn-s3-demo-bucket/<folderName>/",    /* optional */
    },
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new UpdateMetricAttributionCommand(updateMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

En cas de succès, Amazon Personalize renvoie l'ARN de l'attribution métrique que vous avez mise à jour. Pour une liste complète de tous les paramètres, voirUpdateMetricAttribution.

Supprimer une attribution de métrique Amazon Personalize

Si vous ne souhaitez plus générer de rapports, vous pouvez supprimer l'attribution d'une métrique. La suppression d'une attribution de métrique entraîne la suppression de toutes ses métriques et de sa configuration de sortie.

Si vous supprimez l'attribution d'une métrique, Amazon Personalize cesse d'envoyer automatiquement des rapports relatifs à ces données en masse PutEvents et des données incrémentielles à CloudWatch. Les données déjà envoyées CloudWatch ou publiées sur Amazon S3 ne sont pas affectées. Vous pouvez supprimer une attribution de métrique à l'aide de la console Amazon Personalize ou AWS des SDK. AWS Command Line Interface

Rubriques

- Supprimer une attribution métrique (console)
- Suppression d'une attribution métrique (AWS CLI)
- Suppression d'une attribution métrique (AWS SDK)

Supprimer une attribution métrique (console)

Vous supprimez une attribution métrique sur la page d'aperçu de votre attribution métrique.

Pour supprimer une attribution métrique

1. Ouvrez la console Amazon Personalize sur https://console.aws.amazon.com/personalize/homeet connectez-vous à votre compte.
2. Choisissez votre groupe de jeux de données.
3. Dans le volet de navigation, sélectionnez Attribution métrique.
4. Choisissez Supprimer, puis confirmez la suppression.

Suppression d'une attribution métrique (AWS CLI)

Pour supprimer une attribution métrique avec le AWS CLI, utilisez la `delete-metric-attribution` commande suivante.

```
aws personalize delete-metric-attribution --metric-attribution-arn metric attribution
 ARN
```

Suppression d'une attribution métrique (AWS SDK)

Le code suivant montre comment supprimer une attribution de métrique avec le SDK pour Python (Boto3) :

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.delete_metric_attribution(
  metricAttributionArn = 'metric attribution ARN'
)
```

SDK for Java 2.x

```
public static void deleteMetricAttribution(PersonalizeClient client, String
 metricAttributionArn) {

    try {

        DeleteMetricAttributionRequest request =
 DeleteMetricAttributionRequest.builder()
                .metricAttributionArn(metricAttributionArn)
                .build();

        DeleteMetricAttributionResponse response =
 client.deleteMetricAttribution(request);
        if (response.sdkHttpResponse().statusCode() == 200) {
            System.out.println("Metric attribution deleted!");
        }

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { DeleteMetricAttributionCommand, PersonalizeClient } from
```

```
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

export const deleteMetricAttributionParam = {
  metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
};
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new DeleteMetricAttributionCommand(deleteMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Affichage de graphiques de données métriques dans CloudWatch

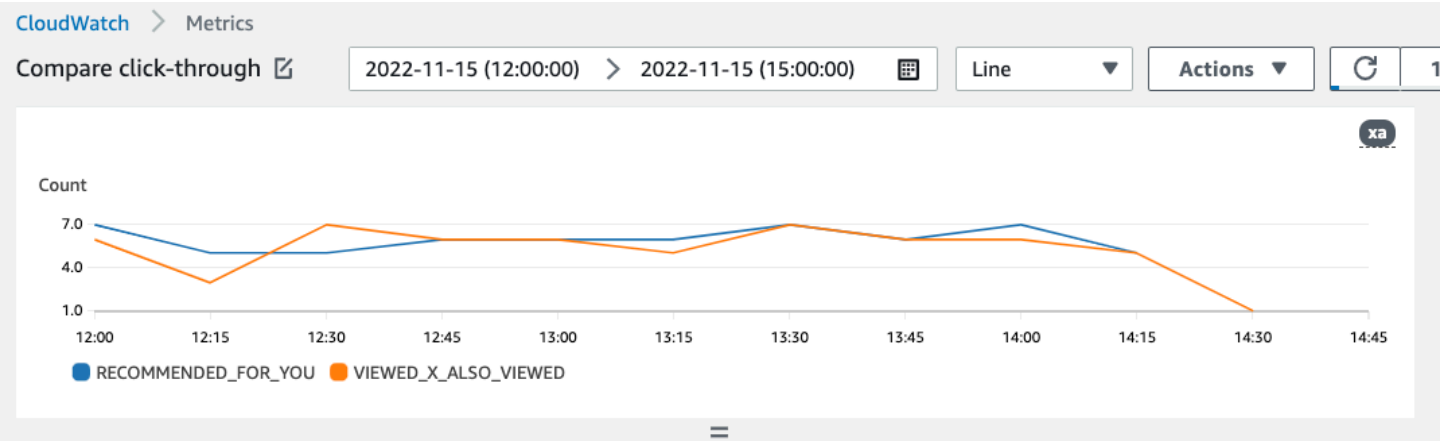
> **Important**
>
> Une fois que vous avez créé une attribution de métrique et enregistré des événements ou importé des données en masse incrémentielles, vous devrez payer un certain CloudWatch coût mensuel par métrique. Pour plus d'informations sur CloudWatch les tarifs, consultez la page de CloudWatch tarification d'Amazon. Pour arrêter d'envoyer des métriques à CloudWatch, supprimez l'attribution des métriques.

Une fois que vous avez créé une attribution de métriques, Amazon Personalize envoie automatiquement des métriques PutEvents et des données en masse incrémentielles à Amazon CloudWatch. Vous pouvez sélectionner des métriques et créer des graphiques des données métriques à l'aide de la CloudWatch console. Ces graphiques peuvent vous aider à inspecter et à comparer visuellement les performances et l'impact de différents recommandateurs ou campagnes.

Pour comparer les sources, chaque événement d'interaction doit inclure un `recommendationId` ou`eventAttributionSource`. Pour des exemples de code montrant comment inclure ces données dans un événement, consultezStatistiques relatives aux événements et rapports d'attribution.

Pour afficher les métriques dans CloudWatch, suivez la procédure décrite dans la section Représentation graphique d'une métrique. Vous pouvez afficher vos données dans différents niveaux de détails. La période minimale que vous pouvez représenter graphiquement est de 15 minutes. Vous pouvez consulter les données Amazon Personalize des 2 semaines précédentes. Les anciennes données sont ignorées. CloudWatch Pour le terme de recherche, spécifiez le nom que vous avez donné à la métrique lorsque vous avez créé l'attribution de la métrique.

Voici un exemple de la façon dont une métrique peut apparaître dans CloudWatch. La métrique indique le taux de clics toutes les 15 minutes pour deux recommandateurs différents.



Publication de rapports d'attribution de métriques sur Amazon S3

Pour toutes les données en masse, si vous fournissez un compartiment Amazon S3 lorsque vous créez votre attribution de métriques, vous pouvez choisir de publier des rapports de métriques dans votre compartiment Amazon S3 chaque fois que vous créez une tâche d'importation d'ensemble de données pour les données d'interactions.

Pour publier des métriques sur Amazon S3, vous indiquez un chemin d'accès à votre compartiment Amazon S3 dans votre attribution de métriques. Vous publiez ensuite des rapports sur Amazon S3 lorsque vous créez une tâche d'importation de jeux de données. Une fois le travail terminé, vous pouvez trouver les métriques dans votre compartiment Amazon S3. Chaque fois que vous publiez des métriques, Amazon Personalize crée un nouveau fichier dans votre compartiment Amazon S3. Le nom du fichier inclut la méthode et la date d'importation, comme suit :

`AggregatedAttributionMetrics - ImportMethod - Timestamp.csv`

Voici un exemple de la façon dont les premières lignes d'un fichier CSV de rapport métrique peuvent apparaître. La métrique présentée dans cet exemple indique le nombre total de clics provenant de deux recommandateurs différents sur des intervalles de 15 minutes. Chaque recommandeur est identifié par son Amazon Resource Name (ARN) dans la colonne EVENT_ATTRIBUTITION_SOURCE.

```
METRIC_NAME,EVENT_TYPE,VALUE,MATH_FUNCTION,EVENT_ATTRIBUTION_SOURCE,TIMESTAMP
COUNTWATCHES,WATCH,12.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender1Name,1666925124
COUNTWATCHES,WATCH,112.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name,1666924224
COUNTWATCHES,WATCH,10.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender1Name,1666924224
COUNTWATCHES,WATCH,254.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name,1666922424
COUNTWATCHES,WATCH,112.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender1Name,1666922424
COUNTWATCHES,WATCH,100.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name,1666922424
......
.....
```

Publication de métriques pour les données en masse sur Amazon S3 (console)

Pour publier des métriques dans un compartiment Amazon S3 à l'aide de la console Amazon Personalize, créez une tâche d'importation de jeux de données et choisissez Publier les métriques pour cette tâche d'importation dans Publier les métriques d'événements sur S3.

Pour obtenir des instructions détaillées, consultez Création d'une tâche d'importation de jeux de données (console).

Publication de métriques pour les données en masse sur Amazon S3 (AWS CLI)

Pour publier des métriques dans un compartiment Amazon S3 avec le AWS Command Line Interface (AWS CLI), utilisez le code suivant pour créer une tâche d'importation de jeu de données et fournir l'`publishAttributionMetricsToS3`indicateur. Si vous ne souhaitez pas publier de statistiques pour une tâche en particulier, omettez l'indicateur. Pour plus d'informations sur chaque paramètre, consultezCreateDatasetImportJob.

```
aws personalize create-dataset-import-job \
--job-name dataset import job name \
--dataset-arn dataset arn \
--data-source dataLocation=s3://amzn-s3-demo-bucket/filename \
--role-arn roleArn \
--import-mode INCREMENTAL \
--publish-attribution-metrics-to-s3
```

Publication de métriques pour les données en masse sur Amazon S3 (AWS SDK)

Pour publier des métriques dans un compartiment Amazon S3 avec les AWS SDK, créez une tâche d'importation de jeux de données et définissez-la sur `publishAttributionMetricsToS3` true. Pour plus d'informations sur chaque paramètre, consultezCreateDatasetImportJob.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation':'s3://amzn-s3-demo-bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'INCREMENTAL',
    publishAttributionMetricsToS3 = True
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
 personalizeClient,
                                                        String jobName,
                                                        String datasetArn,
                                                        String s3BucketPath,
                                                        String roleArn,
                                                        ImportMode importMode,
                                                        boolean publishToS3) {

  long waitInMilliseconds = 60 * 1000;
  String status;
  String datasetImportJobArn;

  try {
      DataSource importDataSource = DataSource.builder()
              .dataLocation(s3BucketPath)
              .build();

      CreateDatasetImportJobRequest createDatasetImportJobRequest =
 CreateDatasetImportJobRequest.builder()
              .datasetArn(datasetArn)
              .dataSource(importDataSource)
              .jobName(jobName)
              .roleArn(roleArn)
              .importMode(importMode)
              .publishAttributionMetricsToS3(publishToS3)
              .build();

      datasetImportJobArn =
 personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
              .datasetImportJobArn();

      DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
 DescribeDatasetImportJobRequest.builder()
              .datasetImportJobArn(datasetImportJobArn)
              .build();

      long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

      while (Instant.now().getEpochSecond() < maxTime) {

          DatasetImportJob datasetImportJob = personalizeClient
```

```
                .describeDatasetImportJob(describeDatasetImportJobRequest)
                .datasetImportJob();

        status = datasetImportJob.status();
        System.out.println("Dataset import job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return datasetImportJobArn;

  } catch (PersonalizeException e) {
      System.out.println(e.awsErrorDetails().errorMessage());
  }
  return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN', /* required */
  dataSource: {
    dataLocation: 's3://amzn-s3-demo-bucket/<folderName>/<CSVfilename>.csv'  /*
 required */
  },
  jobName: 'NAME',                      /* required */
  roleArn: 'ROLE_ARN',                  /* required */
```

```
  importMode: "FULL",                       /* optional, default is FULL */
  publishAttributionMetricsToS3: true     /* set to true to publish metrics to
 Amazon S3 bucket */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
 CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Mesurer l'impact des recommandations à l'aide A/B de tests

La réalisation d'un A/B test consiste à exécuter une expérience avec plusieurs variantes et à comparer les résultats. Pour effectuer A/B des tests avec les recommandations d'Amazon Personalize, il faut présenter différents types de recommandations à différents groupes d'utilisateurs, puis comparer les résultats. Vous pouvez utiliser A/B les tests pour comparer et évaluer différentes stratégies de recommandation, ainsi que pour mesurer l'impact des recommandations.

Par exemple, vous pouvez utiliser des A/B tests pour voir si les recommandations d'Amazon Personalize augmentent le taux de clics. Pour tester ce scénario, vous pouvez afficher à un groupe d'utilisateurs des recommandations qui ne sont pas personnalisées, telles que les produits phares. Et vous pourriez montrer à un autre groupe des recommandations personnalisées générées par Amazon Personalize. Au fur et à mesure que vos clients interagissent avec les articles, vous pouvez enregistrer les résultats et voir quelle stratégie génère le taux de clics le plus élevé.

Le flux de travail pour effectuer A/B des tests avec les recommandations d'Amazon Personalize est le suivant :

1. Planifiez votre expérience : définissez une hypothèse quantifiable, identifiez les objectifs commerciaux, définissez les variantes de l'expérience et déterminez le calendrier de votre expérience.

2. Divisez vos utilisateurs : divisez les utilisateurs en deux groupes ou plus, avec un groupe de contrôle et un ou plusieurs groupes d'expérimentation.
3. Exécutez votre test : affichez les recommandations modifiées aux utilisateurs du groupe d'expérimentation. Afficher les recommandations des utilisateurs du groupe de contrôle sans aucune modification. Enregistrez leurs interactions avec des recommandations pour suivre les résultats.
4. Évaluer les résultats : analysez les résultats de l'expérience pour déterminer si la modification a apporté une différence statistiquement significative pour le groupe d'expérimentation.

Rubriques

- Bonnes pratiques en matière de tests A/B

Bonnes pratiques en matière de tests A/B

Utilisez les meilleures pratiques suivantes pour vous aider à concevoir et à gérer des A/B tests pour les recommandations d'Amazon Personalize.

- Identifiez un objectif commercial quantifiable. Vérifiez que les différentes recommandations que vous souhaitez comparer correspondent à cet objectif commercial et ne sont pas liées à des objectifs différents ou non quantifiables.
- Définissez une hypothèse quantifiable qui correspond à votre objectif commercial. Par exemple, vous pouvez prévoir qu'une promotion pour votre propre contenu personnalisé se traduira par 20 % de clics supplémentaires sur ces articles. Votre hypothèse détermine la modification que vous apportez à votre groupe d'expérimentation.
- Définissez des indicateurs de performance clés pertinents (KPIs) liés à votre hypothèse. Vous l'utilisez KPIs pour mesurer les résultats de vos expériences. Il peut s'agir des suivants :
  - Taux de clics
  - Heure de la montre
  - Prix total
- Vérifiez que le nombre total d'utilisateurs participant à l'expérience est suffisamment important pour obtenir un résultat statistiquement significatif, en fonction de votre hypothèse.
- Définissez votre stratégie de répartition du trafic avant de commencer votre expérience. Évitez de modifier la répartition du trafic pendant que l'expérience est en cours.

- Veillez à ce que l'expérience utilisateur de votre application ou de votre site Web soit la même pour votre groupe d'expérimentation et pour le groupe de contrôle, à l'exception des modifications liées à votre expérience (par exemple, le modèle). Les variations de l'expérience utilisateur, telles que l'interface utilisateur ou la latence, peuvent entraîner des résultats trompeurs.
- Contrôlez les facteurs externes, tels que les jours fériés, les campagnes marketing en cours et les limitations du navigateur. Ces facteurs externes peuvent entraîner des résultats trompeurs.
- Évitez de modifier les recommandations d'Amazon Personalize, sauf si elles sont directement liées à votre hypothèse ou à des exigences commerciales. Des modifications telles que l'application d'un filtre ou la modification manuelle de l'ordre peuvent entraîner des résultats trompeurs.
- Lorsque vous évaluez les résultats, assurez-vous qu'ils sont statistiquement significatifs avant de tirer des conclusions. La norme de l'industrie est un seuil de signification de 5 %.

Personnalisation des résultats de recherche à partir de OpenSearch

Vous pouvez utiliser Amazon Personalize pour personnaliser les résultats de l'open source OpenSearch ou d'Amazon OpenSearch Service pour vos utilisateurs.

OpenSearchest un service de recherche open source autogéré basé sur la licence Apache 2.0. Amazon OpenSearch Service est un service géré qui vous aide à déployer, exploiter et dimensionner les OpenSearch ressources dans le AWS cloud. Lorsque vous utilisez Amazon OpenSearch Service, OpenSearch récupère et classe les résultats.

Lors du classement des résultats d'une requête, OpenSearch utilise un cadre de classement probabiliste appelé BM-25 pour calculer les scores de pertinence. Si un mot clé distinctif apparaît plus fréquemment dans un document, BM-25 attribue un score de pertinence plus élevé à ce document. OpenSearch le classement ne prend pas en compte le comportement des utilisateurs, comme les données relatives aux clics.

Lorsque vous utilisez Amazon Personalize with OpenSearch, Amazon Personalize reclasse les OpenSearch résultats en fonction du comportement passé de l'utilisateur, de toutes les métadonnées relatives aux articles et de toutes les métadonnées concernant l'utilisateur. OpenSearch intègre ensuite le reclassement avant de renvoyer la réponse de recherche à votre application. Vous contrôlez le poids accordé OpenSearch au classement d'Amazon Personalize lorsque vous l'appliquez aux OpenSearch résultats.

Grâce à ce reclassement, les résultats peuvent être plus intéressants et plus pertinents par rapport aux intérêts de l'utilisateur. Cela peut entraîner une augmentation du taux de clics et du taux de conversion de votre application. Pour un exemple de cas d'utilisation décrivant comment la recherche personnalisée peut améliorer les résultats d'une application de commerce électronique, voirExemple de cas d'utilisation.

Avant de commencer à personnaliser les OpenSearch résultats, passez en revue les exigences répertoriées dansExigences relatives au plugin Amazon Personalize Search Ranking.

Rubriques

- Exemple de cas d'utilisation
- Comment fonctionne le plugin Amazon Personalize Search Ranking

- Informations supplémentaires
- Exigences relatives au plugin Amazon Personalize Search Ranking
- Personnalisation des résultats d'Amazon OpenSearch Service avec Amazon Personalize
- Personnalisation des résultats depuis Open Source Open Search avec Amazon Personalize
- Champs pour le processeur personalized_search_ranking de réponse
- Exemple de métriques de pipeline

Exemple de cas d'utilisation

Lorsque vous utilisez Amazon Personalize pour reclasser les OpenSearch résultats, les résultats de recherche peuvent être plus pertinents pour vos utilisateurs. Par exemple, vous pourriez avoir une application de commerce électronique qui vend des voitures. Si votre utilisateur saisit une requête pour des voitures Toyota et que vous ne personnalisez pas les résultats, OpenSearch il renverra une liste de voitures fabriquées par Toyota en fonction des mots clés contenus dans vos données. Cette liste serait classée dans le même ordre pour tous les utilisateurs.

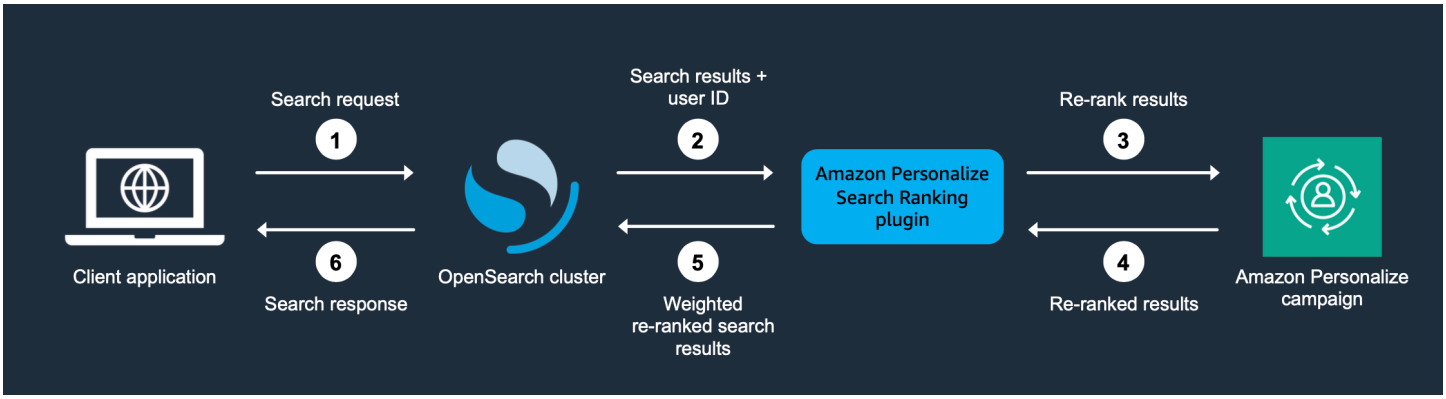
Mais si vous utilisez Amazon Personalize pour personnaliser les résultats, OpenSearch reclassez ces voitures par ordre de pertinence pour chaque utilisateur en fonction de son comportement, par exemple ses clics. La voiture sur laquelle l'utilisateur est le plus susceptible de cliquer est classée première.

Lorsque vous personnalisez OpenSearch les résultats, vous contrôlez le poids (accent) accordé OpenSearch au classement d'Amazon Personalize. En reprenant cet exemple, si un utilisateur recherche un type de voiture spécifique pour une année donnée (comme une Toyota Prius 2008), vous souhaiterez peut-être mettre davantage l'accent sur le classement original de OpenSearch.

Toutefois, pour les requêtes plus génériques qui donnent lieu à un large éventail de résultats (comme une recherche pour tous les véhicules Toyota), vous pouvez mettre l'accent sur la personnalisation. Ainsi, les voitures figurant en haut de la liste sont plus pertinentes pour l'utilisateur en question.

Comment fonctionne le plugin Amazon Personalize Search Ranking

Le schéma suivant montre le fonctionnement du plugin Amazon Personalize Search Ranking.



1. Vous soumettez la requête de votre client à votre domaine OpenSearch de service ou à votre OpenSearch cluster open source.
2. OpenSearch envoie la réponse à la requête (liste des éléments pertinents pour la requête) et l'identifiant de l'utilisateur au plugin Amazon Personalize Search Ranking.
3. Le plugin envoie les articles et l'utilisateur en réponse à votre campagne Amazon Personalize à des fins de classement. Il utilise les valeurs ARN (Amazon Resource Name) de la recette et de la campagne dans votre pipeline de recherche pour obtenir un classement personnalisé pour l'utilisateur. Il utilise le fonctionnement de GetPersonalizedRanking l'API pour les recommandations. Dans la demande, il transmet l'userID de l'utilisateur effectuant la requête et les éléments renvoyés par la OpenSearch requête dans le. `inputList`
4. Amazon Personalize renvoie les résultats reclassés au plugin.
5. Le plugin réorganise et renvoie les résultats de recherche à votre domaine de OpenSearch service ou à votre cluster open source OpenSearch . Il reclasse les résultats en fonction de la réponse de votre campagne Amazon Personalize et de l'accent mis sur la personnalisation que vous spécifiez lors de la configuration.
6. Votre OpenSearch cluster open source ou votre domaine de OpenSearch service renvoie les résultats finaux à votre application.

Informations supplémentaires

Les ressources suivantes fournissent des informations supplémentaires sur l'utilisation OpenSearch.

- Pour plus d'informations sur la prise en main de l'open source OpenSearch, consultez Quickstart.
- Pour plus d'informations sur la prise en OpenSearch main de Service, consultez Getting started with Amazon OpenSearch Service dans le manuel Amazon OpenSearch Service Developer Guide.

- Pour plus d'informations sur les recettes de classement personnalisé dans Amazon Personalize, consultez ou. Recette de classement V2 personnalisée Recette de classement personnalisé

Exigences relatives au plugin Amazon Personalize Search Ranking

Avant de commencer à personnaliser les résultats à partir de OpenSearch, prenez note des directives et exigences suivantes concernant le plugin Amazon Personalize Search Ranking :

- Vous devez utiliser OpenSearch la version 2.9.0 ou ultérieure. Si vous utilisez Amazon OpenSearch Service, votre domaine doit utiliser la version 2.9 ou ultérieure.
- Si ce n'est pas déjà fait, suivez les instructions Configurer les autorisations pour autoriser vos utilisateurs à accéder à Amazon Personalize et autoriser Amazon Personalize à accéder à vos ressources dans Amazon Personalize.
- Vous devez être en mesure d'accéder à vos ressources Amazon Personalize depuis votre domaine de OpenSearch service ou votre OpenSearch cluster open source.
  - Pour plus d'informations sur l'octroi de l'accès à un domaine de OpenSearch service, consultezConfiguration des autorisations Amazon OpenSearch Service.
  - Pour plus d'informations sur l'octroi d'accès à un OpenSearch cluster, consultezConfiguration des OpenSearch autorisations open source.
- Vous ne pouvez utiliser que des ressources Amazon Personalize personnalisées. Si vous avez créé un groupe de jeux de données de domaine, vous pouvez toujours ajouter des ressources personnalisées.
- Vous ne pouvez utiliser que les recettes personnalisées Recette de classement V2 personnalisée ouRecette de classement personnalisé.
- Vous devez créer un ensemble de données sur les interactions entre articles dans Amazon Personalize. Les ensembles de données des éléments et des utilisateurs sont facultatifs.
- Vous ne pouvez pas appliquer de filtres Amazon Personalize lorsque vous utilisez le plugin Amazon Personalize Search Ranking.
- Par défaut, le plugin part du principe que le nom `_id` d'un document indexé OpenSearch correspond à l'itemID indiqué dans vos données Amazon Personalize. Si vos OpenSearch données utilisent un champ différent correspondant à vos Amazon Personalize ItemIds, vous devez spécifier le nom du champ lorsque vous configurez le plugin.
- L'UserID que vous utilisez pour un utilisateur effectuant une requête doit correspondre à son UserID dans les données que vous importez dans Amazon Personalize.

- Le plugin ne reclasse que les 500 premiers résultats de recherche de OpenSearch. Les objets restants ne sont pas reclassés et se retrouvent en bas de la liste.

Personnalisation des résultats d'Amazon OpenSearch Service avec Amazon Personalize

Pour personnaliser OpenSearch les résultats d'Amazon OpenSearch Service, procédez comme suit :

1. Configurer Amazon Personalize — Si ce n'est pas déjà fait, suivez les étapes décrites Configuration d'Amazon Personalize pour configurer vos informations d'identification et les autorisations pour Amazon Personalize. Vous n'avez pas besoin de configurer le AWS SDKs pour personnaliser les OpenSearch résultats.
2. Terminez le flux de travail Amazon Personalize — Complétez le flux de travail Amazon Personalize pour importer des données, créer une solution avec la recette Personalized-Ranking-v 2 ou Personalized-Ranking, former une version de solution personnalisée et la déployer dans le cadre d'une campagne. Vous devez créer un jeu de données sur les interactions entre les articles. Un ensemble de données Users et un ensemble de données Items sont facultatifs. Pour de plus amples informations, veuillez consulter Détails du flux de travail Amazon Personalize.
3. Configurer des autorisations : configurez des autorisations afin de pouvoir accéder à vos ressources Amazon Personalize depuis votre domaine de OpenSearch service. Pour de plus amples informations, veuillez consulter Configurer les autorisations.
4. Installez le plugin Amazon Personalize Search Ranking : ce plugin gère la communication avec Amazon Personalize et le reclassement des résultats. Pour plus d'informations sur l'installation du plug-in sur un domaine de OpenSearch service, consultezInstallation du plug-in.
5. Configurer le plugin Amazon Personalize Search Ranking : pour configurer le plugin, vous devez créer des pipelines de recherche. Les pipelines de recherche sont des ensembles de processeurs de requêtes et de réponses. Lorsque vous créez un pipeline pour le plugin, vous spécifiez vos ressources Amazon Personalize dans un processeur de `personalized_search_ranking` réponse. Vous configurez également le poids que le plugin attribue aux résultats d'Amazon Personalize lorsqu'il reclasse les résultats. Pour de plus amples informations, veuillez consulter Création d'un pipeline.
6. Appliquez le plugin Amazon Personalize Search Ranking aux OpenSearch requêtes : après avoir configuré un pipeline de recherche avec un processeur de `personalized_search_ranking` réponse, vous êtes prêt à appliquer le plug-in Amazon Personalize Search Ranking à vos

OpenSearch requêtes et à consulter les résultats reclassés. Pour plus d'informations sur l'application du plug-in aux requêtes de OpenSearch service, consultezAppliquer le plugin.

7. Comparer les résultats — Le plugin Amazon Personalize Search Ranking reclasse les résultats de recherche dans la réponse à la OpenSearch requête. Il prend en compte à la fois le classement d'Amazon Personalize et le classement de OpenSearch. Pour comprendre comment les résultats sont reclassés, vous pouvez comparer les résultats des requêtes qui utilisent la personnalisation et ceux qui ne le font pas. Pour plus d'informations sur la comparaison des résultats avec le OpenSearch Service, consultezComparaison des résultats.
8. Surveillez le plugin Amazon Personalize Search Ranking — Lorsque vous appliquez le plug-in Amazon Personalize Search Ranking aux requêtes de recherche, vous pouvez surveiller le plug-in en obtenant des statistiques pour vos pipelines de recherche. Pour plus d'informations sur la surveillance du plugin avec OpenSearch Service, consultezSurveillance du plugin.

Rubriques

- Configuration des autorisations Amazon OpenSearch Service
- Installation du plugin Amazon Personalize Search Ranking sur un domaine OpenSearch de service
- Création d'un pipeline dans Amazon OpenSearch Service
- Appliquer le plugin aux requêtes Amazon OpenSearch Service
- Comparaison des résultats personnalisés d'Amazon OpenSearch Service avec des résultats sans personnalisation
- Surveillance du plugin avec Amazon OpenSearch Service

Configuration des autorisations Amazon OpenSearch Service

Si vous utilisez Amazon OpenSearch Service, vous devez être en mesure d'accéder à vos ressources Amazon Personalize depuis votre domaine de OpenSearch service.

Pour configurer les autorisations

1. Selon que vos ressources se trouvent dans le même compte ou dans des comptes différents, créez un ou plusieurs rôles de service IAM autorisés à accéder à vos ressources.
   - Si vos ressources OpenSearch Service et Amazon Personalize se trouvent sur le même compte, vous créez un rôle de service IAM pour OpenSearch Service et vous lui accordez l'autorisation d'obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Pour

de plus amples informations, veuillez consulter Configuration des autorisations lorsque les ressources se trouvent dans le même compte.

- Si vos ressources OpenSearch Service et Amazon Personalize se trouvent dans des comptes distincts, vous créez deux rôles de service IAM. Vous en créez un dans le compte avec vos ressources de OpenSearch service et vous lui accordez l'accès à vos ressources OpenSearch de service. Et vous en créez une dans le compte avec vos ressources Amazon Personalize et vous lui accordez l'autorisation d'obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Pour de plus amples informations, veuillez consulter Configuration des autorisations lorsque les ressources se trouvent dans des comptes différents.

2. Accordez à l'utilisateur ou au rôle qui accède à votre domaine de OpenSearch service `PassRole` des autorisations pour le rôle de service IAM que vous avez créé pour OpenSearch Service. Pour de plus amples informations, veuillez consulter Configuration de la sécurité OpenSearch du domaine Amazon Service.

Après avoir configuré les autorisations, vous êtes prêt à installer le plugin sur votre domaine. Pour de plus amples informations, veuillez consulter Installation du plug-in.

Rubriques

- Configuration des autorisations lorsque les ressources se trouvent dans le même compte
- Configuration des autorisations lorsque les ressources se trouvent dans des comptes différents
- Configuration de la sécurité OpenSearch du domaine Amazon Service

Configuration des autorisations lorsque les ressources se trouvent dans le même compte

Si vos ressources OpenSearch Service et Amazon Personalize se trouvent dans le même compte, vous devez créer un rôle de service IAM pour OpenSearch Service. Ce rôle doit être autorisé à obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Les conditions suivantes sont requises pour autoriser votre rôle de OpenSearch service dans le service à obtenir un classement personnalisé à partir de votre campagne Amazon Personalize :

- La politique de confiance du rôle doit accorder `AssumeRole` des autorisations pour le OpenSearch service. Pour un exemple de politique de confiance, voirExemple de politique de confiance.
- Le rôle doit être autorisé à obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Pour un exemple de stratégie, consultez Exemple de politique d'autorisations.

Pour plus d'informations sur la création d'un rôle IAM, consultez la section Création de rôles IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'attachement d'une politique IAM à un rôle, consultez la section Ajouter et supprimer des autorisations d'identité IAM dans le guide de l'utilisateur IAM.

Après avoir créé un rôle de service IAM pour le OpenSearch service, vous devez accorder à l'utilisateur ou au rôle qui accède à votre domaine de OpenSearch service `PassRole` des autorisations pour le rôle de OpenSearch service de service. Pour de plus amples informations, veuillez consulter Configuration de la sécurité OpenSearch du domaine Amazon Service.

Rubriques

- Exemple de politique de confiance
- Exemple de politique d'autorisations

Exemple de politique de confiance

L'exemple de politique de confiance suivant accorde des `AssumeRole` autorisations pour le OpenSearch service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Principal": {
            "Service": [
                "es.amazonaws.com"
            ]
        }
    }]
}
```

Exemple de politique d'autorisations

L'exemple de politique suivant accorde au rôle les autorisations minimales nécessaires pour obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Pour`Campaign ARN`, spécifiez le nom de ressource Amazon (ARN) de votre campagne Amazon Personalize.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

Configuration des autorisations lorsque les ressources se trouvent dans des comptes différents

Si vos ressources OpenSearch Service et Amazon Personalize se trouvent dans des comptes distincts, vous créez un rôle IAM dans chaque compte et vous accordez au rôle l'accès aux ressources du compte.

Pour configurer des autorisations pour plusieurs comptes

1. Dans le compte sur lequel existe votre campagne Amazon Personalize, créez un rôle IAM autorisé à obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Lorsque vous configurez le plug-in, vous spécifiez l'ARN pour ce rôle dans le `external_account_iam_role_arn` paramètre du processeur de `personalized_search_ranking` réponse. Pour de plus amples informations, veuillez consulter Création d'un pipeline dans Amazon OpenSearch Service.

   Pour un exemple de stratégie, consultez Exemple de politique d'autorisations.

2. Dans le compte où se trouve votre domaine de OpenSearch service, créez un rôle doté d'une politique de confiance qui accorde des `AssumeRole` autorisations OpenSearch de service. Lorsque vous configurez le plug-in, vous spécifiez l'ARN pour ce rôle dans le `iam_role_arn` paramètre du processeur de `personalized_search_ranking` réponse. Pour de plus amples informations, veuillez consulter Création d'un pipeline dans Amazon OpenSearch Service.

   Pour un exemple de politique de confiance, voirExemple de politique de confiance.

3. Modifiez chaque rôle pour accorder des `AssumeRole` autorisations à l'autre rôle. Par exemple, pour le rôle qui a accès à vos ressources Amazon Personalize, sa politique IAM accordera au rôle dans le compte avec le domaine OpenSearch Service les autorisations d'assumer le rôle comme suit :

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [{
           "Sid": "",
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::111122223333:role/roleName"

       }]
   }
   ```

4. Dans le compte où se trouve votre domaine de OpenSearch service, accordez à l'utilisateur ou au rôle qui accède à votre domaine de OpenSearch service `PassRole` des autorisations pour le rôle de OpenSearch service de service que vous venez de créer. Pour de plus amples informations, veuillez consulter Configuration de la sécurité OpenSearch du domaine Amazon Service.

Configuration de la sécurité OpenSearch du domaine Amazon Service

Pour utiliser le plug-in avec OpenSearch Service, l'utilisateur ou le rôle qui accède à votre domaine doit disposer d'`PassRole`autorisations pour le rôle de service IAM pour le OpenSearch service que vous venez de créer. En outre, l'utilisateur ou le rôle doit être autorisé à effectuer les `es:ESHttpPut` actions `es:ESHttpGet` et.

Pour plus d'informations sur la configuration de l'accès au OpenSearch service, consultez la section Sécurité d'Amazon OpenSearch Service dans le manuel Amazon OpenSearch Service Developer Guide. Pour obtenir des exemples de stratégie, consultez Exemples de politiques pour l'utilisateur ou le rôle du OpenSearch service.

Exemples de politiques pour l'utilisateur ou le rôle du OpenSearch service

L'exemple de politique IAM suivant accorde à un utilisateur ou à un rôle `PassRole` des autorisations pour le rôle de service IAM que vous avez créé pour OpenSearch Service in. Configuration des autorisations lorsque les ressources se trouvent dans le même compte

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService"
        }
    ]
}
```

La politique IAM suivante accorde les autorisations minimales pour créer des pipelines et des requêtes de recherche avec OpenSearch Service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:ESHttpGet",
                "es:ESHttpPut"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:ResourceTag/environment": [
                        "production"
                    ]
                }
            }
        }
    ]
}
```

Installation du plugin Amazon Personalize Search Ranking sur un domaine OpenSearch de service

Une fois que vous avez terminé le flux de travail Amazon Personalize et que vous avez satisfait aux exigences répertoriées dans le documentExigences relatives aux plugins, vous êtes prêt à installer le plug-in sur votre domaine.

Pour utiliser le plugin, vous devez l'`Amazon_Personalize_Search_Ranking_Plugin`associer à votre domaine. Le plugin est préinstallé et vous n'avez pas besoin de l'importer depuis Amazon S3. Vous associez le plugin de la même manière que vous associez un OpenSearch Service package. Pour plus d'informations sur l'association d'un pack de OpenSearch services, consultez la section Packages personnalisés pour Amazon OpenSearch Service.

Après avoir associé le plugin à votre domaine, vous êtes prêt à le configurer. Vous le configurez en créant un pipeline de recherche et en spécifiant un processeur de `personalized_search_ranking` réponse. Pour de plus amples informations, veuillez consulter Création d'un pipeline.

Informations supplémentaires sur les domaines Amazon OpenSearch Service

Les ressources suivantes fournissent des informations supplémentaires sur l'utilisation du domaine Amazon OpenSearch Service.

- Pour un didacticiel concis sur la configuration d'un domaine de test, consultez Étape 1 : Création d'un domaine Amazon OpenSearch Service dans la section « Getting started » du manuel Amazon OpenSearch Service Developer Guide.

- Pour des étapes plus détaillées sur la configuration des domaines de OpenSearch service, consultez Création et gestion des domaines Amazon OpenSearch Service.
- Pour un didacticiel concis sur le téléchargement d'une petite quantité de données de test vers OpenSearch Service, consultez l'étape 2 : télécharger des données vers Amazon OpenSearch Service pour les indexer dans la section « Getting started » du manuel Amazon OpenSearch Service Developer Guide.
- Pour des informations complètes sur l'ingestion de données, consultez la section Indexation des données dans Amazon OpenSearch Service dans le manuel Amazon OpenSearch Service Developer Guide.

Création d'un pipeline dans Amazon OpenSearch Service

Après avoir installé le plugin Amazon Personalize Search Ranking, vous êtes prêt à le configurer en créant un pipeline OpenSearch de recherche.

Un pipeline de recherche est un ensemble de processeurs de requêtes et de réponses qui s'exécutent de manière séquentielle dans l'ordre dans lequel vous les avez créés. Lorsque vous créez un pipeline de recherche pour le plugin, vous spécifiez un processeur de `personalized_search_ranking` réponse. Pour plus d'informations sur les pipelines de recherche, consultez la section Pipelines de recherche.

Après avoir créé un pipeline de recherche avec un processeur de `personalized_search_ranking` réponse, vous êtes prêt à appliquer le plugin aux OpenSearch requêtes. Vous pouvez l'appliquer à un OpenSearch index ou à une OpenSearch requête individuelle. Pour de plus amples informations, veuillez consulter Appliquer le plugin.

Vous pouvez utiliser le code Python suivant pour créer un pipeline de recherche avec un processeur de `personalized_search_ranking` réponse sur un domaine OpenSearch de service. Remplacez par `domain endpoint` l'URL du point de terminaison de votre domaine. Par exemple : `https://<domain name>.<AWS region>.es-staging.amazonaws.com`. Pour une explication complète de chaque `personalized_search_ranking` paramètre, voirChamps pour le processeur `personalized_search_ranking` de réponse.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
pipeline_name = 'pipeline name'
```

```
url = f'{domain_endpoint}/_search/pipeline/{pipeline_name}'
auth = AWSSigV4('es')

headers = {'Content-Type': 'application/json'}

body = {
  "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
  "response_processors": [
    {
      "personalized_search_ranking" : {
        "campaign_arn" : "Amazon Personalize Campaign ARN",
        "item_id_field" : "productId",
        "recipe" : "aws-personalized-ranking-v2",
        "weight" : "0.3",
        "tag" : "personalize-processor",
        "iam_role_arn": "Role ARN",
        "aws_region": "AWS region",
        "ignore_failure": true
    }
  ]
}
try:
    response = requests.put(url, auth=auth, json=body, headers=headers, verify=False)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

Appliquer le plugin aux requêtes Amazon OpenSearch Service

Après avoir créé un pipeline, vous êtes prêt à appliquer le plugin Amazon Personalize Search Ranking aux requêtes. Vous pouvez appliquer le plugin Amazon Personalize Search Ranking à toutes les requêtes et réponses relatives à un index. Vous pouvez également appliquer le plugin à des requêtes et réponses individuelles.

- Vous pouvez utiliser le code Python suivant pour appliquer un pipeline de recherche à un index. Avec cette approche, toutes les recherches utilisant cet index utilisent le plugin pour appliquer la personnalisation aux résultats de recherche.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
```

```
index = 'index name'
url = f'{domain_endpoint}/{index}/_settings/'
auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
body = {
    "index.search.default_pipeline": "pipeline name"
}
try:
    response = requests.put(url, auth=auth, json=body, headers=headers)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

- Vous pouvez utiliser le code Python suivant pour appliquer un pipeline de recherche à une requête individuelle pour des voitures de marque Toyota.

  Mettez à jour le code pour spécifier le point de terminaison de votre domaine, votre index de OpenSearch service, le nom de votre pipeline et votre requête. Pour`user_id`, spécifiez l'ID de l'utilisateur pour lequel vous obtenez des résultats de recherche. Cet utilisateur doit figurer dans les données que vous avez utilisées pour créer la version de votre solution Amazon Personalize. Si l'utilisateur n'était pas présent, Amazon Personalize classe les articles en fonction de leur popularité.

  En `context` effet, si vous utilisez des métadonnées contextuelles, fournissez les métadonnées contextuelles de l'utilisateur, telles que son type d'appareil. Le champ `context` est facultatif. Pour de plus amples informations, veuillez consulter Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles.

  ```
  import requests
  from requests_auth_aws_sigv4 import AWSSigV4

  domain_endpoint = 'domain endpoint'
  index = 'index name'
  url = f'{domain_endpoint}/{index}/_search/'

  auth = AWSSigV4('es')
  headers = {'Content-Type': 'application/json'}
  params = {"search_pipeline": "pipeline-name"}
  body = {
      "query": {
          "multi_match": {
              "query": "Toyota",
  ```

```
            "fields": ["BRAND"]
        }
    },
    "ext": {
        "personalize_request_parameters": {
            "user_id": "USER ID",
            "context": { "DEVICE" : "mobile phone" }
        }
    }
}
try:
    response = requests.post(url, auth=auth, params=params, json=body,
 headers=headers, verify=False)
    print(response)
except Exception as e:
    print(f"Error: {e}")
```

Comparaison des résultats personnalisés d'Amazon OpenSearch Service avec des résultats sans personnalisation

Le plugin Amazon Personalize Search Ranking réorganise les résultats de recherche en fonction à la fois du classement d'Amazon Personalize et du classement de. OpenSearch La façon dont le plugin reclasse les résultats dépend de la façon dont vous avez configuré le processeur de `personalized_search_ranking` réponse dans vos pipelines.

Pour comprendre le classement des résultats, vous pouvez exécuter des requêtes avec ou sans personnalisation et comparer les résultats. Vous pouvez utiliser le code Python suivant pour exécuter deux requêtes différentes et générer les résultats dans deux fichiers JSON. La première méthode exécute une requête qui utilise le plugin pour reclasser les résultats. Le second exécute une méthode qui génère des résultats sans personnalisation.

```
import json
import requests
from requests_auth_aws_sigv4 import AWSSigV4


# Returns re-ranked OpenSearch results using the Amazon Personalize Search Ranking
 plugin.
def get_personalized_results(pipeline_name):
    url = f'{domain}/{index}/_search/'
```

```
    auth = AWSSigV4('es')
    headers = {'Content-Type': 'application/json'}
    params = {"search_pipeline": pipeline_name}
    body = {
        "query": {
            "multi_match": {
                "query": "Toyota",
                "fields": ["BRAND"]
            }
        },
        "ext": {
            "personalize_request_parameters": {
                "user_id": "1"
            }
        }
    }
    try:
        response = requests.post(url, auth=auth, params=params, json=body,
 headers=headers, verify=False)
    except Exception as e:
        return f"Error: {e}"
    return response.text


# Returns OpenSearch results without personalization.
def get_opensearch_results():
    url = f'{domain}/{index}/_search/'
    auth = AWSSigV4('es')
    headers = {'Content-Type': 'application/json'}
    body = {
        "query": {
            "multi_match": {
                "query": "Toyota",
                "fields": ["BRAND"]
            }
        }
    }
    try:
        response = requests.post(url, auth=auth, json=body, headers=headers,
 verify=False)
    except Exception as e:
        return f"Error: {e}"
    return response.text
```

```
def print_results(file_name, results):
    results_file = open(file_name, 'w')
    results_file.write(json.dumps(results, indent=4))
    results_file.close()


# specify domain endpoint
domain = "DOMAIN_ENDPOINT"

# specify the region where you created your Amazon Personalize resources and Amazon
 OpenSearch domain
aws_region = "REGION"

# specify the name of the pipeline that uses the Amazon Personalize plugin
pipeline_name = "PIPELINE_NAME"

# specify your Amazon OpenSearch index
index = "INDEX"

# specify names for json files for comparison
personalized_results_file = "personalized_results.json"
opensearch_results_file = "opensearch_results.json"

# get personalized results
personalized_results = json.loads(get_personalized_results(pipeline_name))

# get OpenSearch results without personalization
opensearch_results = json.loads(get_opensearch_results())

# print results to files
print_results(personalized_results_file, personalized_results)
print_results(opensearch_results_file, opensearch_results)
```

Surveillance du plugin avec Amazon OpenSearch Service

Lorsque vous appliquez le plugin Amazon Personalize Search Ranking aux OpenSearch requêtes, vous pouvez surveiller le plug-in en obtenant des statistiques pour vos pipelines de recherche. Les métriques du pipeline incluent des statistiques telles que le nombre de demandes ayant échoué pour le processeur de `personalized_search_ranking` réponse.

Si vous utilisez le OpenSearch service, vous pouvez surveiller le plugin via des métriques sur Amazon CloudWatch. Pour plus d'informations, consultez la section Surveillance des domaines Amazon OpenSearch Service.

Vous pouvez utiliser le code Python suivant pour obtenir des métriques pour tous vos pipelines. Pour un exemple de métriques de pipeline, voirExemple de métriques de pipeline.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
url = f'{domain_endpoint}/_nodes/stats/search_pipeline'

auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
try:
    response = requests.get(url, auth=auth, headers=headers, verify=False)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

Personnalisation des résultats depuis Open Source Open Search avec Amazon Personalize

Pour personnaliser les résultats en open source OpenSearch, procédez comme suit :

1. Configurer Amazon Personalize — Si ce n'est pas déjà fait, suivez les étapes décrites Configuration d'Amazon Personalize pour configurer vos informations d'identification et les autorisations pour Amazon Personalize. Vous n'avez pas besoin de configurer le AWS SDKs pour personnaliser les OpenSearch résultats.
2. Terminez le flux de travail Amazon Personalize — Complétez le flux de travail Amazon Personalize pour importer des données, créer une solution avec la recette Personalized-Ranking-v 2 ou Personalized-Ranking, former une version de solution personnalisée et la déployer dans le cadre d'une campagne. Vous devez créer un jeu de données sur les interactions entre les articles. Un ensemble de données Users et un ensemble de données Items sont facultatifs. Pour de plus amples informations, veuillez consulter Détails du flux de travail Amazon Personalize.

3. Configurer des autorisations : configurez des autorisations afin de pouvoir accéder à vos ressources Amazon Personalize depuis votre OpenSearch cluster. Pour de plus amples informations, veuillez consulter Configurer les autorisations.
4. Installez le plugin Amazon Personalize Search Ranking — Le plugin Amazon Personalize Search Ranking gère la communication avec Amazon Personalize et le reclassement des résultats.
   - Si un OpenSearch cluster est déjà en cours d'exécution, vous pouvez installer le plugin manuellement. Pour de plus amples informations, veuillez consulter Installation manuelle du plugin sur un OpenSearch cluster existant.
   - Si vous n'avez pas créé de OpenSearch cluster, vous pouvez utiliser un script bash de démarrage rapide pour en créer un. Pour de plus amples informations, veuillez consulter Création d'un nouveau cluster et installation du plugin avec un script.
5. Configurer le plugin Amazon Personalize Search Ranking : pour configurer le plugin, vous devez créer des pipelines de recherche. Les pipelines de recherche sont des ensembles de processeurs de requêtes et de réponses. Lorsque vous créez un pipeline pour le plugin, vous spécifiez vos ressources Amazon Personalize dans un processeur de `personalized_search_ranking` réponse. Vous configurez également le poids que le plugin attribue aux résultats d'Amazon Personalize lorsqu'il reclasse les résultats. Pour de plus amples informations, veuillez consulter Création d'un pipeline.
6. Appliquer le plugin Amazon Personalize Search Ranking aux OpenSearch requêtes : vous pouvez appliquer le plug-in Amazon Personalize Search Ranking à toutes les requêtes et réponses d'un OpenSearch index. Vous pouvez également appliquer le plugin à des OpenSearch requêtes et réponses individuelles. Pour plus d'informations sur l'application du plugin aux requêtes en open source OpenSearch, consultezAppliquer le plugin.
7. Comparer les résultats — Le plugin Amazon Personalize Search Ranking reclasse les résultats de recherche dans la réponse à la OpenSearch requête. Il prend en compte à la fois le classement d'Amazon Personalize et le classement de OpenSearch. Pour comprendre comment les résultats sont reclassés, vous pouvez comparer les résultats des requêtes qui utilisent la personnalisation et ceux qui ne le font pas. Pour plus d'informations sur la comparaison des résultats avec l'open source OpenSearch, consultezComparaison des résultats.
8. Surveillez le plugin Amazon Personalize Search Ranking — Lorsque vous appliquez le plug-in Amazon Personalize Search Ranking aux requêtes de recherche, vous pouvez surveiller le plug-in en obtenant des statistiques pour vos pipelines de recherche. Pour plus d'informations sur la surveillance du plug-in sur un OpenSearch cluster open source, consultezSurveillance du plugin avec l'open source OpenSearch. Pour un extrait des métriques du pipeline renvoyées OpenSearch, voirExemple de métriques de pipeline.

Rubriques

- Configuration des OpenSearch autorisations open source
- Installation manuelle du plugin Amazon Personalize Search Ranking sur un OpenSearch cluster existant
- Création d'un nouveau cluster et installation du plugin avec un script
- Création d'un pipeline en open source OpenSearch
- Appliquer le plugin Amazon Personalize Search Ranking aux requêtes en open source OpenSearch
- Comparaison de OpenSearch résultats personnalisés à des résultats sans personnalisation
- Surveillance du plugin avec l'open source OpenSearch

Configuration des OpenSearch autorisations open source

Si vous utilisez l'open source OpenSearch, vous devez être en mesure d'accéder à vos ressources Amazon Personalize depuis votre cluster de recherche ouvert. Pour accorder l'accès, procédez comme suit :

- Si vous effectuez une configuration à OpenSearch partir de zéro, vous pouvez utiliser un script bash de démarrage rapide pour exécuter un OpenSearch cluster dans un conteneur Docker. Le script utilise les informations d'identification par défaut de votre AWS profil. Vous pouvez spécifier un autre profil lorsque vous exécutez le script.

  Ces informations d'identification doivent être associées à un utilisateur ou à un rôle autorisé à effectuer l' GetPersonalizedRankingaction associée à votre campagne Amazon Personalize. Pour un exemple de stratégie IAM, consultezExemples de politique IAM. Les informations d'identification doivent également être autorisées à assumer un rôle doté de ces autorisations. Vous pouvez fournir l'Amazon Resource Name (ARN) pour ce rôle lorsque vous créez un pipeline pour le plugin Amazon Personalize Search Ranking.
- Si vous n'utilisez pas le script bash de démarrage rapide, vous pouvez ajouter manuellement vos informations d'identification à votre OpenSearch keystore. Ces informations d'identification doivent correspondre à un utilisateur ou à un rôle autorisé à effectuer l' GetPersonalizedRanking action pour votre campagne Amazon Personalize.

  Pour ajouter manuellement vos AWS informations d'identification à votre OpenSearch keystore, exécutez la commande suivante là où votre OpenSearch cluster est exécuté (par exemple, un

conteneur Docker). Fournissez ensuite chaque identifiant. Si vous n'utilisez pas de jeton de session, vous pouvez omettre la dernière ligne de la commande.

```
opensearch-keystore add \
personalized_search_ranking.aws.access_key \
personalized_search_ranking.aws.secret_key \
personalized_search_ranking.aws.session_token
```

- Si vous exécutez votre OpenSearch cluster sur une instance Amazon EC2, vous pouvez accorder des autorisations avec un profil d'instance IAM. La politique associée au rôle doit l'autoriser à effectuer l' GetPersonalizedRanking action correspondant à votre campagne Amazon Personalize. Il doit également accorder à Amazon EC2 les autorisations nécessaires pour assumer ce rôle.

  Pour plus d'informations sur les profils d'instance Amazon EC2, consultez la section Utilisation des profils d'instance. Pour un exemple de stratégie, consultez Exemples de politique IAM.

Exemples de politique IAM

L'exemple de politique suivant accorde à un utilisateur ou à un rôle les autorisations minimales nécessaires pour obtenir un classement personnalisé à partir de votre campagne Amazon Personalize. Pour`Campaign ARN`, spécifiez le nom de ressource Amazon (ARN) de votre campagne Amazon Personalize.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

En outre, si vous exécutez votre OpenSearch cluster sur une instance Amazon EC2 et que vous accordez des autorisations avec un profil d'instance IAM, la politique de confiance relative au rôle doit accorder les autorisations Amazon `AssumeRole` EC2 comme suit. Pour plus d'informations sur les profils d'instance Amazon EC2, consultez la section Utilisation des profils d'instance.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Installation manuelle du plugin Amazon Personalize Search Ranking sur un OpenSearch cluster existant

Si vous possédez déjà un OpenSearch cluster, vous pouvez installer manuellement le plugin sur votre cluster directement depuis le OpenSearch GitHub référentiel.

Pour installer manuellement le plugin

1. Utilisez la commande suivante pour démarrer votre OpenSearch cluster :

   ```
   bin/opensearch
   ```

2. Si ce n'est pas déjà fait, téléchargez les données de votre catalogue sur votre OpenSearch cluster. Lorsque vous téléchargez vos données, vous créez un OpenSearch index et définissez vos mappages de champs. Ensuite, vous chargez vos données dans cet index. Pour un exemple, voir Création d'un index et mappages de champs à l'aide d'exemples de données.
3. Utilisez la commande suivante pour installer le plugin :

```
bin/opensearch-plugin install https://github.com/opensearch-project/search-
processor/releases/download/2.9.0/opensearch-search-processor-2.9.0.0.zip
```

Pour plus d'informations sur l'installation de plug-ins, consultez la section Installation de plug-ins.

Après avoir installé le plugin Amazon Personalize Search Ranking, vous êtes prêt à le configurer. Vous configurez le plugin en créant un pipeline de recherche et en spécifiant un processeur de `personalized_search_ranking` réponse. Pour de plus amples informations, veuillez consulter Création d'un pipeline.

Création d'un nouveau cluster et installation du plugin avec un script

Si vous n'avez pas créé de OpenSearch cluster, vous pouvez utiliser un script bash de démarrage rapide pour en créer un. Ce script configure un OpenSearch cluster dans un conteneur Docker, définit les informations d'identification à l'aide de votre AWS profil par défaut et installe le plugin Amazon Personalize Search Ranking.

Pour plus d'informations sur la création manuelle d'un OpenSearch cluster, consultez les instructions de démarrage rapide de la OpenSearch documentation.

Pour installer le plugin avec un script bash de démarrage rapide

1. Avant d'exécuter le script, téléchargez et installez Docker Desktop pour votre système d'exploitation.
2. Téléchargez le script bash de démarrage rapide depuis GitHub.
3. Dans votre répertoire de travail, exécutez le script avec la commande suivante.

   ```
   sh personalized_search_ranking_quickstart.sh
   ```

   Avec cette commande, le script utilise les informations d'identification de votre AWS profil par défaut. Pour fournir un autre profil, utilisez l'`--profile`argument.

   ```
   sh personalized_search_ranking_quickstart.sh --profile profile-name
   ```

   Après avoir exécuté le script, vous trouverez plus d'informations sur le script dans le fichier README situé dans le répertoire unique créé par le script. Ce répertoire contient les fichiers

Dockerfile et docker-compose.yml utilisés par le script. Par exemple : `../opensearch-personalize-intelligent-ranking-docker.1234/README`.

4. Téléchargez les données de votre catalogue sur votre OpenSearch cluster. Lorsque vous téléchargez vos données, vous créez un OpenSearch index et définissez vos mappages de champs. Vous chargez ensuite vos données dans cet index. Pour un exemple, voir Création d'un index et mappages de champs à l'aide d'exemples de données.

Après avoir configuré OpenSearch et installé le plugin Amazon Personalize Search Ranking, vous êtes prêt à le configurer. Vous configurez le plugin en créant un pipeline de recherche et en spécifiant un processeur de `personalized_search_ranking` réponse. Pour de plus amples informations, veuillez consulter Création d'un pipeline.

Création d'un pipeline en open source OpenSearch

Après avoir installé le plugin sur votre cluster, vous êtes prêt à le configurer en créant un pipeline OpenSearch de recherche.

Un pipeline de recherche est un ensemble de processeurs de requêtes et de réponses qui s'exécutent de manière séquentielle dans l'ordre dans lequel vous les avez créés. Lorsque vous créez un pipeline de recherche pour le plugin, vous spécifiez un processeur de `personalized_search_ranking` réponse. Pour plus d'informations sur les pipelines de recherche, consultez la section Pipelines de recherche.

Après avoir créé un pipeline avec un processeur de `personalized_search_ranking` réponse, vous êtes prêt à appliquer le plugin aux requêtes. Pour de plus amples informations, veuillez consulter Appliquer le plugin.

Vous pouvez utiliser la commande curl suivante pour créer un pipeline de recherche avec un processeur de `personalized_search_ranking` réponse sur un OpenSearch cluster open source. Pour une explication complète de chaque `personalized_search_ranking` paramètre, voirChamps pour le processeur `personalized_search_ranking` de réponse.

```
curl -X PUT "http://localhost:9200/_search/pipeline/pipeline-name" -ku 'admin:admin' --
insecure -H 'Content-Type: application/json' -d'
{
  "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
  "response_processors" : [
    {
```

```
      "personalized_search_ranking" : {
        "campaign_arn" : "Amazon Personalize Campaign ARN",
        "item_id_field" : "productId",
        "recipe" : "aws-personalized-ranking-v2",
        "weight" : "0.3",
        "tag" : "personalize-processor",
        "iam_role_arn": "Role ARN",
        "aws_region": "AWS region",
        "ignore_failure": true
      }
    }
  ]
}'
```

Après avoir créé un pipeline de recherche avec un processeur de `personalized_search_ranking` réponse, vous êtes prêt à appliquer le plugin aux OpenSearch requêtes. Vous pouvez l'appliquer à un OpenSearch index ou à une OpenSearch requête individuelle. Pour de plus amples informations, veuillez consulter Appliquer le plugin Amazon Personalize Search Ranking aux requêtes en open source OpenSearch.

Appliquer le plugin Amazon Personalize Search Ranking aux requêtes en open source OpenSearch

Vous pouvez appliquer le plugin Amazon Personalize Search Ranking à toutes les requêtes et réponses relatives à un OpenSearch index. Vous pouvez également appliquer le plugin à des OpenSearch requêtes et réponses individuelles.

- La commande curl suivante applique un pipeline de recherche à un OpenSearch index dans un OpenSearch cluster open source exécuté localement. Avec cette approche, toutes les recherches effectuées dans cet index utilisent le plugin pour appliquer la personnalisation aux résultats de recherche.

  ```
  curl -XGET "https://localhost:9200/index/_settings" -ku 'admin:admin' --insecure -H
   'Content-Type: application/json' -d'
  {
    "index.search.default_pipeline": "pipeline-name"
  }
  '
  ```

- La commande curl suivante applique un pipeline de recherche à une requête individuelle pour des voitures de marque Toyota sur un index d'un OpenSearch cluster open source exécuté localement.

Pour`user_id`, spécifiez l'ID de l'utilisateur pour lequel vous obtenez des résultats de recherche. Cet utilisateur doit figurer dans les données que vous avez utilisées pour créer la version de votre solution Amazon Personalize. Si l'utilisateur n'était pas présent, Amazon Personalize classe les articles en fonction de leur popularité. En `context` effet, si vous utilisez des métadonnées contextuelles, fournissez les métadonnées contextuelles de l'utilisateur, telles que son type d'appareil. Le champ `context` est facultatif. Pour de plus amples informations, veuillez consulter Améliorer la pertinence des recommandations grâce aux métadonnées contextuelles.

```
curl -XGET "http://localhost:9200/index/_search?search_pipeline=pipeline-name" -ku
 'admin:admin' --insecure -H 'Content-Type: application/json' -d'
{
  "query": {
    "multi_match": {
      "query": "Toyota",
      "fields": ["BRAND"]
    }
  },
  "ext": {
    "personalize_request_parameters": {
      "user_id": "USER ID",
      "context": { "DEVICE": "mobile phone" }
    }
  }
}
'
```

Pour comprendre comment les résultats sont reclassés, vous pouvez utiliser les OpenSearch tableaux de bord pour comparer les résultats aux OpenSearch résultats reclassés avec le plugin. Pour de plus amples informations, veuillez consulter Comparaison de OpenSearch résultats personnalisés à des résultats sans personnalisation.

Lorsque vous appliquez le plugin aux OpenSearch requêtes, vous pouvez le surveiller en obtenant des métriques pour votre OpenSearch pipeline. Pour de plus amples informations, veuillez consulter Surveillance du plugin avec l'open source OpenSearch.

Comparaison de OpenSearch résultats personnalisés à des résultats sans personnalisation

Pour comprendre comment les résultats sont reclassés, vous pouvez exécuter des requêtes à l'aide de la console Dev Tools dans deux fenêtres de navigateur distinctes. Vous pouvez ensuite comparer les résultats des requêtes avec et sans personnalisation.

Pour comparer les résultats avec la console Dev Tools

1. Assurez-vous que OpenSearch Dashboards est installé. Le script bash Quickstart installe OpenSearch les tableaux de bord. Si vous n'utilisez pas le script ou si un cluster est déjà en cours d'exécution, vous devez installer OpenSearch Dashboards. Pour plus d'informations, consultez la section Installation de OpenSearch tableaux de bord.
2. Lancez les OpenSearch tableaux de bord. Ouvrez `http://localhost:5601` depuis un navigateur et connectez-vous aux OpenSearch tableaux de bord. Les informations d'identification par défaut sont le nom d'utilisateur « admin » et le mot de passe « admin ».
3. Choisissez Dev Tools dans le menu Management de la page d'accueil OpenSearch des tableaux de bord.
4. Ouvrez une fenêtre de navigateur séparée et ouvrez à nouveau la console Dev Tools. Vous pouvez utiliser l'URL de la fenêtre précédente.
5. Dans une fenêtre, saisissez une requête qui n'utilise aucun reclassement à des fins de personnalisation. Dans l'autre fenêtre, entrez une commande curl qui utilise un pipeline avec le processeur de `personalized_search_ranking` réponse. Si vous collez une commande curl directement dans la console, elle est automatiquement convertie dans le format utilisé par la console. Pour un exemple de commande, voirAppliquer le plugin Amazon Personalize Search Ranking aux requêtes en open source OpenSearch.
6. Exécutez les deux requêtes et comparez les résultats.

Surveillance du plugin avec l'open source OpenSearch

Lorsque vous appliquez le plugin Amazon Personalize Search Ranking aux OpenSearch requêtes, vous pouvez surveiller le plug-in en obtenant des statistiques pour vos pipelines de recherche. Les métriques du pipeline incluent des statistiques telles que le nombre de demandes ayant échoué pour le processeur de `personalized_search_ranking` réponse.

Vous pouvez utiliser le code suivant pour obtenir des métriques pour tous vos pipelines. La réponse contient des statistiques pour tous les pipelines de recherche. Pour un exemple de métriques de pipeline, voirExemple de métriques de pipeline.

```
curl -XGET "https://localhost:9200/_nodes/stats/search_pipeline?pretty" -ku
 'admin:admin'
```

Champs pour le processeur **personalized_search_ranking** de réponse

Lorsque vous créez un pipeline de recherche pour le plugin Amazon Personalize Search Ranking, vous spécifiez un processeur de `personalized_search_ranking` réponse avec les champs suivants.

- campaign_arn (obligatoire) — Spécifiez le nom de ressource Amazon (ARN) de la campagne Amazon Personalize à utiliser pour personnaliser les résultats.
- item_id_field (facultatif) — Si le `_id` champ d'un document indexé dans OpenSearch ne correspond pas à vos Amazon Personalize ItemID, spécifiez le nom du champ correspondant. Par défaut, le plugin part du principe que les `_id` données correspondent à l'itemID de vos données Amazon Personalize.
- recette (obligatoire) — Spécifiez le nom de la recette Amazon Personalize à utiliser. Pour utiliser le plugin, vous pouvez spécifier `aws-personalized-ranking-v2` ou`aws-personalized-ranking`.
- poids (obligatoire) — Spécifiez l'accent que le processeur de réponse met sur la personnalisation lorsqu'il reclasse les résultats. Spécifiez une valeur comprise entre 0 et 1,0. Plus `1.0` on s'en rapproche, plus il est probable que les résultats d'Amazon Personalize soient mieux classés. Si vous le spécifiez`0.0`, aucune personnalisation n'est effectuée et OpenSearch est prioritaire.
- tag (facultatif) — Spécifiez un identifiant pour le processeur.
- iam_role_arn (obligatoire pour le OpenSearch service, facultatif pour l'open source OpenSearch) : pour le service OpenSearch , indiquez le nom de ressource Amazon (ARN) pour le rôle que vous avez créé lors de la configuration des autorisations permettant à Service d' OpenSearch accéder à vos ressources Amazon Personalize. Si vos ressources OpenSearch Service et Amazon Personalize existent sur des comptes différents, spécifiez le rôle qui accorde `AssumeRole` les autorisations pour le OpenSearch Service. Pour de plus amples informations, veuillez consulter Configuration des autorisations lorsque les ressources se trouvent dans des comptes différents.

Pour l'open source OpenSearch, si vous utilisez plusieurs rôles pour restreindre les autorisations de différents groupes d'utilisateurs de votre organisation, spécifiez l'ARN du rôle autorisé à accéder à Amazon Personalize. Si vous utilisez uniquement les AWS informations d'identification de votre OpenSearch keystore, vous pouvez omettre ce champ.

- aws_region (obligatoire) : AWS région dans laquelle vous avez créé votre campagne Amazon Personalize.
- ignore_failure (facultatif) — Spécifiez si le plugin ignore les défaillances du processeur. Pour les valeurs, spécifiez `true` ou`false`. Pour vos environnements de production, nous vous recommandons de spécifier `true` afin d'éviter toute interruption des réponses aux requêtes. Pour les environnements de test, vous pouvez `false` spécifier d'afficher les erreurs générées par le plugin.
- external_account_iam_role_arn — Si vous utilisez Service et que vos ressources Amazon Personalize OpenSearch et Service existent sur des comptes différents, spécifiez l'ARN du rôle autorisé à accéder à vos OpenSearch ressources Amazon Personalize. Ce rôle doit exister dans le même compte que vos ressources Amazon Personalize. Pour de plus amples informations, veuillez consulter Configuration des autorisations lorsque les ressources se trouvent dans des comptes différents.

Pour un exemple OpenSearch de code de service, voirCréation d'un pipeline dans Amazon OpenSearch Service. Pour un OpenSearch exemple de code source ouvert, voirCréation d'un pipeline en open source OpenSearch.

Exemple de métriques de pipeline

Lorsque vous appliquez le plugin Amazon Personalize Search Ranking aux OpenSearch requêtes, vous pouvez surveiller le plug-in en obtenant des statistiques pour vos pipelines de recherche. Les métriques du pipeline incluent des statistiques telles que le nombre de demandes ayant échoué pour le processeur de `personalized_search_ranking` réponse.

Le code suivant montre un extrait des métriques du pipeline renvoyées par OpenSearch. Il affiche uniquement l'`pipelines`objet contenant les statistiques de deux pipelines différents. Pour chaque pipeline, vous pouvez trouver les statistiques du plugin Amazon Personalize Search Ranking dans la liste des processeurs de `personalized_search_ranking` réponse. Pour un exemple complet de toutes les métriques, consultez la section Métriques du pipeline de recherche.

```
{
```

```
....
....
  "pipelines": {
    "pipelineA": {
      "request": {
        "count": 0,
        "time_in_millis": 0,
        "current": 0,
        "failed": 0
      },
      "response": {
        "count": 6,
        "time_in_millis": 2246,
        "current": 0,
        "failed": 0
      },
      "request_processors": [],
      "response_processors": [
        {
          personalized_search_ranking": {
            "type": "personalized_search_ranking",
            "stats": {
              "count": <number of requests>,
              "time_in_millis": <time>,
              "current": 0,
              "failed": <number of failed requests>
            }
          }
        }
      ]
    },
    "pipelineB": {
      "request": {
        "count": 0,
        "time_in_millis": 0,
        "current": 0,
        "failed": 0
      },
      "response": {
        "count": 8,
        "time_in_millis": 2248,
        "current": 0,
        "failed": 0
      },
```

```
      "request_processors": [],
      "response_processors": [
        {
          "personalized_search_ranking": {
            "type": "personalized_search_ranking",
            "stats": {
              "count": <number of requests>,
              "time_in_millis": <time>,
              "current": 0,
              "failed": <number of failed requests>
            }
          }
        }
      ]
    }
  }
....
....
}
```

Marquer les ressources Amazon Personalize

Une balise est une étiquette que vous pouvez éventuellement définir et associer à AWS des ressources, notamment à certains types de ressources Amazon Personalize. Une ressource peut avoir jusqu'à 50 balises.

Les balises peuvent vous aider à classer et à gérer les ressources de différentes manières, par exemple en fonction de leur objectif, de leur environnement ou d'autres critères. Par exemple, vous pouvez utiliser des balises pour répartir les revenus entre différentes fonctions ou identifier des environnements de développement pour différentes ressources.

Pour récupérer les ressources Amazon Personalize par balise, vous pouvez utiliser les filtres utilisés par l' GetResources API Resource Groups Tagging. Pour plus d'informations, consultez le guide GetResourcesde référence de l'API Resource Groups Tagging API.

Vous pouvez ajouter des balises aux types de ressources Amazon Personalize suivants :

- Tâches d’inférence par lots
- Tâches liées au segment Batch
- Campagnes
- Ensembles de données
- Groupes de jeux de données
- Tâches d'importation et d'exportation de données
- Traqueurs d'événements
- Filtres
- Recommandeurs
- Des solutions
- Versions des solutions

Rubriques

- Directives et exigences
- Ajouter des balises aux ressources Amazon Personalize
- Supprimer les tags des ressources Amazon Personalize
- Utilisation de balises dans les politiques IAM

Directives et exigences

Chaque balise est constituée d'une clé de balise obligatoire et d'une valeur de balise facultative que vous définissez. Une clé de balise est une étiquette générale qui fait office de catégorie pour des valeurs de balise plus spécifiques. Une valeur de balise tient lieu de descripteur pour une clé de balise.

Par exemple, si vous disposez de deux versions d'un groupe de données Amazon Personalize (l'une pour les tests internes et l'autre pour la production), vous pouvez attribuer une clé de `Environment` balise aux deux projets. La valeur de balise de la `Environment` balise peut correspondre `Test` à une version du groupe de jeux de données et `Production` à l'autre version.

Les restrictions suivantes s'appliquent aux balises :

- Nombre maximal d'étiquettes par ressource : 50
- Longueur de clé maximale : 128 caractères Unicode en UTF-8
- Longueur de valeur maximale : 256 caractères Unicode en UTF-8
- Les clés et les valeurs des balises peuvent contenir les caractères suivants : A-Z, a-z, 0-9, espace et _. :/= + @ — (tiret). Il s'agit du jeu de caractères standard disponible dans tous les AWS services qui prennent en charge les balises. Certains services prennent en charge des symboles supplémentaires.
- Les clés de balise et valeurs de balise sont sensibles à la casse.
- Pour chaque ressource associée, chaque clé de balise doit être unique et ne peut avoir qu'une seule valeur de balise.
- Les clés et les valeurs de vos balises ne peuvent pas commencer par`aws:`. AWS les services appliquent des balises qui commencent par`aws:`, et ces balises ne peuvent pas être modifiées. Ils ne sont pas pris en compte dans les limites de tags.
- Vous ne pouvez pas mettre à jour ou supprimer une ressource uniquement en fonction de ses balises. Vous devez également spécifier l'Amazon Resource Name (ARN) ou l'ID de ressource selon l'opération que vous utilisez.

Informations supplémentaires

Pour plus d'informations sur le balisage, consultez les ressources suivantes.

- AWS Principes de balisage dans le manuel de référence AWS général

- AWS Stratégies de balisage (PDF téléchargeable)
- AWS Le contrôle d'accès dans le guide de l'utilisateur AWS IAM
- AWS Politiques de balisage dans le guide de l'AWS Organizations utilisateur

Ajouter des balises aux ressources Amazon Personalize

Vous pouvez ajouter, afficher, mettre à jour et supprimer des clés et des valeurs de balise dans les ressources Amazon Personalize à l'aide de la console Amazon Personalize, AWS Command Line Interface (AWS CLI), ou AWS SDKs. Les exemples suivants montrent comment ajouter une balise au groupe de jeux de données Amazon Personalize. Vous pouvez ajouter des balises à d'autres ressources Amazon Personalize de la même manière.

Rubriques

- Ajouter des tags (console)
- Ajouter des tags (AWS CLI)
- Ajouter des tags (AWS SDKs)

Ajouter des tags (console)

Lorsque vous créez une ressource dans Amazon Personalize, vous pouvez ajouter des balises facultatives à l'aide de la console Amazon Personalize. L'exemple suivant ajoute une balise à un groupe de jeux de données.

Pour ajouter des balises à un nouveau groupe de jeux de données

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez Create dataset group (Créer un groupe d'ensembles de données).
3. Pour Nom, entrez un nom.
4. Pour Domaine, choisissez un domaine.
5. Développez la section Tags et choisissez Ajouter un nouveau tag.
6. Pour Clé et Valeur, entrez les valeurs appropriées.

   Par exemple, respectivement, **`Environment`** et **`Test`**.

7. Pour ajouter d'autres balises, choisissez Ajouter une nouvelle étiquette.

   Vous pouvez ajouter jusqu'à 50 balises à une ressource.
8. Choisissez Next pour continuer à créer votre ressource.

L'ajout de balises à une ressource existante est similaire : choisissez votre ressource et utilisez les champs Tags pour ajouter vos balises.

Ajouter des tags (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour ajouter des balises lorsque vous créez une ressource ou ajoutez des balises à une ressource existante.

Rubriques

- Ajouter des balises lors de la création d'une ressource
- Ajout de balises à une ressource existante

Ajouter des balises lors de la création d'une ressource

Pour créer une nouvelle ressource et y ajouter une balise avec le AWS CLI, utilisez la `create` commande appropriée pour la ressource et incluez le `tags` paramètre et les valeurs. Par exemple, la commande suivante crée un nouveau groupe de jeux de données de domaine nommé `myDatasetGroup` d'après le domaine ECOMMERCE et ajoute les balises suivantes : une clé de `Environment` balise avec une valeur de `Test` balise, et une clé de `Owner` balise et une `xyzCorp` valeur.

```
aws personalize create-dataset-group \
--name myDatasetGroup \
--domain ECOMMERCE \
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

Pour plus d'informations sur les commandes que vous pouvez utiliser pour créer une ressource Amazon Personalize, consultez le manuel Amazon Personalize AWS CLI Command Reference.

Ajout de balises à une ressource existante

Pour ajouter une balise à une ressource existante, utilisez la `tag-resource` commande. Spécifiez l'ARN de la ressource et indiquez la clé de balise et la valeur dans le `tags` paramètre.

```
aws personalize tag-resource \
--resource-arn resource ARN \
--tags tagKey=key,tagValue=value
```

Ajouter des tags (AWS SDKs)

Vous pouvez utiliser le AWS SDKs pour ajouter des balises lorsque vous créez une ressource ou pour ajouter des balises à une ressource existante.

Rubriques

- Ajouter des balises lors de la création d'une ressource
- Ajout de balises à une ressource existante

Ajouter des balises lors de la création d'une ressource

Pour créer une nouvelle ressource et y ajouter une étiquette avec le AWS SDKs, utilisez la `create` méthode appropriée. Utilisez le `tags` paramètre pour spécifier les paires clé-valeur pour chacune de vos balises. Par exemple, le code suivant crée un nouveau groupe de jeux de données de domaine nommé `myDatasetGroup` d'après le domaine ECOMMERCE et ajoute les balises suivantes : une clé de `Environment` balise avec une valeur de `Test` balise, et une clé de `Owner` balise et une `xyzCorp` valeur.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
  name = 'myDatasetGroup',
  domain = 'ECOMMERCE',
  tags = [
    {
      'tagKey': 'Environment',
      'tagValue': 'Test'
    },
    {
      'tagKey': 'Owner',
      'tagValue': 'xyzCorp'
```

```
    }
  ]
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                               String datasetGroupName,
                                               String domain) {

    try {

        ArrayList <Tag> tags = new ArrayList<>();

        Tag tag1 = Tag.builder()
                .tagKey("Environment")
                .tagValue("Test")
                .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
                .tagKey("Owner")
                .tagValue("xyzCorp")
                .build();
        tags.add(tag2);

        CreateDatasetGroupRequest createDatasetGroupRequest =
 CreateDatasetGroupRequest.builder()
                .name(datasetGroupName)
                .domain(domain)
                .tags(tags)
                .build();
        return
 personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
```

```
    }
    return "";
}
```

Ajout de balises à une ressource existante

Le code suivant montre comment ajouter une balise à une ressource Amazon Personalize existante. Spécifiez le nom de ressource Amazon (ARN) de la ressource à laquelle vous souhaitez ajouter des balises et spécifiez des paires clé-valeur pour chacune de vos balises.

SDK for Python (Boto3)

```
import boto3
personalize = boto3.client('personalize')

add_tags_response = personalize.tag_resource(
  resourceArn = "resourceArn",
  tags = [
    {
      'tagKey': 'Environment',
      'tagValue': 'Test'
    },
    {
      'tagKey': 'Owner',
      'tagValue': 'xyzCorp'
    }
  ]
)
```

SDK for Java 2.x

```
public static void tagResource(PersonalizeClient personalizeClient,
                                            String resourceArn,
                                            String domain) {

    try {

         ArrayList <Tag> tagList = new ArrayList<>();

          Tag tag1 = Tag.builder()
                  .tagKey("Environment")
```

```
                .tagValue("Test")
                .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
                .tagKey("Owner")
                .tagValue("xyzCorp")
                .build();
        tags.add(tag2);

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                .resourceArn(resourceArn)
                .tags(tagList)
                .build();

        personalizeClient.tagResource(tagResourceRequest);
        System.out.println("Tags have been added to "+ resourceArn);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

Supprimer les tags des ressources Amazon Personalize

Vous pouvez supprimer les balises des ressources Amazon Personalize à l'aide de la console Amazon Personalize ou à l'aide de l'UntagResourceAPI AWS Command Line Interface (AWS CLI) ou AWS SDKs. Les exemples suivants montrent comment supprimer une balise d'un groupe de jeux de données Amazon Personalize. Vous pouvez supprimer les tags d'autres ressources Amazon Personalize de la même manière.

Rubriques

- Supprimer des tags (console)
- Supprimer les tags (AWS CLI)
- Supprimer les tags (AWS SDKs)

Supprimer des tags (console)

Après avoir ajouté des balises à une ressource dans Amazon Personalize, vous pouvez les supprimer à l'aide de la console Amazon Personalize. L'exemple suivant supprime une balise d'un groupe de jeux de données.

Pour supprimer des balises d'un groupe de jeux de données

1. Ouvrez la console Amazon Personalize https://console.aws.amazon.com/personalize/chez vous et connectez-vous à votre compte.
2. Choisissez votre groupe de jeux de données.
3. Au bas de la page, cliquez sur l'onglet Tags, puis sur Gérer les tags.
4. Pour chaque étiquette que vous souhaitez supprimer, choisissez Supprimer.
5. Choisissez Enregistrer pour supprimer les balises.

Supprimer les tags (AWS CLI)

Pour supprimer des balises d'une ressource existante avec le AWS CLI, utilisez la `untag-resource` commande suivante. Pour`resource-arn`, spécifiez le Amazon Resource Name (ARN) de la ressource. Pour`tag-keys`, spécifiez les clés des balises à supprimer.

```
aws personalize untag-resource \
--resource-arn resource ARN \
--tag-keys key1 key2
```

Supprimer les tags (AWS SDKs)

Pour supprimer des balises d'une ressource Amazon Personalize existante à l'aide du AWS SDKs, utilisez l'opération UntagResource API. Le code suivant montre comment supprimer plusieurs balises d'un groupe de jeux de données à l'aide du SDK pour Python (Boto3). Pour`resourceArn`, spécifiez le Amazon Resource Name (ARN) de la ressource. Pour`tagKeys`, spécifiez les clés des balises à supprimer.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.untag_resource(
```

```
    resourceArn="Resource ARN",
    tagKeys=["tag1Key", "tag2Key"]
)
```

Utilisation de balises dans les politiques IAM

Après avoir commencé à implémenter des balises, vous pouvez appliquer des autorisations de niveau ressource basées sur des balises aux politiques Gestion des identités et des accès AWS (IAM) et aux opérations d'API. Cela inclut les opérations qui prennent en charge l'ajout de balises aux ressources lors de la création de ressources. En utilisant les balises de cette manière, vous pouvez contrôler de manière précise quels groupes et utilisateurs de votre AWS compte sont autorisés à créer et à étiqueter des ressources, et quels groupes et utilisateurs sont autorisés à créer, mettre à jour et supprimer des balises de manière plus générale.

Par exemple, vous pouvez créer une politique qui permet à un utilisateur d'avoir un accès complet à toutes les ressources Amazon Personalize dont le nom est une valeur dans le `Owner` tag de la ressource.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "personalize:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

L'exemple suivant montre comment créer une politique permettant de créer et de supprimer un ensemble de données. Ces opérations ne sont autorisées que si le nom d'utilisateur est`johndoe`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:CreateDataset",
                "personalize:DeleteDataset"
            ],
            "Resource": "arn:aws:personalize:*:*:dataset/*",
            "Condition": {
                "StringEquals": {"aws:username" : "johndoe"}
            }
        },
        {
            "Effect": "Allow",
            "Action": "personalize:DescribeDataset",
            "Resource": "*"
        }
    ]
}
```

Si vous définissez des autorisations au niveau des ressources basées sur des balises, les autorisations prennent effet immédiatement. Vos ressources sont ainsi plus sécurisées dès leur création et vous pouvez rapidement commencer à appliquer l'utilisation des balises pour les nouvelles ressources. Vous pouvez également utiliser des autorisations au niveau des ressources afin de contrôler les clés et les valeurs de balise qui peuvent être associés à des ressources nouvelles et existantes. Pour plus d'informations, consultez Contrôle de l'accès à l'aide de balises dans le Guide de l'utilisateur AWS IAM.

Questions fréquemment posées sur Amazon Personalize

Vous trouverez ci-dessous les réponses aux questions fréquemment posées concernant l'importation de données, la formation, le déploiement de modèles, les recommandations et les filtres dans Amazon Personalize.

Pour plus de questions et réponses, consultez l'aide-mémoire Amazon Personalize dans le référentiel d'exemples Amazon Personalize.

Rubriques

- Importation et gestion des données
- Création d'une solution personnalisée et d'une version de solution
- Déploiement de modèles (campagnes personnalisées)
- Recommandations
- Recommandations de filtrage

Importation et gestion des données

Dans quel format doivent être mes données groupées ?

Vos données groupées doivent être au format CSV (valeurs séparées par des virgules). La première ligne de votre fichier CSV doit contenir des en-têtes de colonne. Les en-têtes de colonne de votre fichier CSV doivent correspondre au schéma utilisé pour créer l'ensemble de données. Si vos données contiennent des caractères non ASCII, votre fichier CSV doit être codé au format UTF-8. Ne placez pas les en-têtes entre guillemets («). `TIMESTAMP`et `CREATION_TIMESTAMP` les données doivent être au format Epoch Time Unix. Pour plus d'informations sur les données d'horodatage, consultez. Données d'horodatage Pour plus d’informations sur les schémas, consultez Création de fichiers JSON pour les schémas Amazon Personalize.

Pour consulter les directives complètes relatives au format des données, voirPréparation des données de formation pour Amazon Personalize. Si vous ne savez pas comment formater vos données, vous pouvez utiliser Amazon SageMaker AI Data Wrangler (Data Wrangler) pour les préparer. Pour de plus amples informations, veuillez consulter Préparation et importation de données en masse à l'aide d'Amazon SageMaker AI Data Wrangler.

De combien de données d'entraînement ai-je besoin ?

Pour tous les cas d'utilisation (groupes de jeux de données de domaine) et les recettes personnalisées, vos données d'interactions doivent présenter les éléments suivants :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour des recommandations de qualité, nous vous recommandons d'avoir au moins 50 000 interactions entre articles provenant d'au moins 1 000 utilisateurs, avec au moins deux interactions avec des articles chacun.

Vous pouvez commencer avec un jeu de données d'interactions Item vide et, lorsque vous avez enregistré suffisamment de données, créer votre recommandation (groupe de jeux de données de domaine) ou une version de solution personnalisée en utilisant uniquement les nouveaux événements enregistrés. Certaines recettes et certains cas d'utilisation peuvent nécessiter des données supplémentaires. Pour plus d'informations sur les exigences relatives aux cas d'utilisation, consultezChoix d'un cas d'utilisation. Pour plus d'informations sur les exigences relatives aux recettes, voirChoisir une recette.

Comment mettre à jour les attributs d'un article ou d'un utilisateur ?

Utilisez la console Amazon Personalize PutItems ou les PutUsers opérations or pour importer un article ou un utilisateur avec le même identifiant d'article mais avec les attributs modifiés.

Comment supprimer un élément ou un utilisateur ?

Amazon Personalize ne prend pas en charge la suppression d'un article ou d'un utilisateur spécifique. Pour vous assurer qu'un élément ou un utilisateur n'apparaît pas dans les recommandations, utilisez un filtre pour exclure des éléments. Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Comment supprimer un schéma ?

Vous ne pouvez supprimer un schéma qu'à l'aide de DeleteSchema cette opération. Vous ne pouvez pas utiliser la console Amazon Personalize pour supprimer un schéma.

Création d'une solution personnalisée et d'une version de solution

Quelle recette dois-je utiliser ?

La recette Amazon Personalize que vous utilisez dépend de votre cas d'utilisation. Pour plus d'informations sur la correspondance entre les cas d'utilisation et les recettes, consultezChoisir une recette. L'aide-mémoire Amazon Personalize inclut également des informations sur les cas d'utilisation et les recettes.

À quelle fréquence dois-je m'entraîner ?

Nous recommandons d'utiliser un entraînement automatique avec une fréquence d'entraînement d'au moins une semaine. La formation automatique vous permet de maintenir plus facilement la pertinence des recommandations. La fréquence de vos formations dépend des besoins de votre entreprise, de la recette que vous utilisez et de la fréquence à laquelle vous importez des données. Pour de plus amples informations, veuillez consulter Configuration de l'entraînement automatique. Pour plus d'informations sur le maintien de la pertinence, voirMaintien de la pertinence des recommandations.

Dois-je utiliser AutoML ?

Non, nous vous recommandons plutôt de faire correspondre votre cas d'utilisation aux différentes recettes Amazon Personalize et de choisir une recette. Pour plus d'informations sur la correspondance entre les cas d'utilisation et les recettes, consultezChoisir une recette.

Déploiement de modèles (campagnes personnalisées)

Que dois-je définir pour le MinProvisionedTPS de ma campagne ?

Un montant élevé `minProvisionedTPS` augmentera vos coûts. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire.

Comment puis-je contrôler le coût de mes campagnes ?

Le projet Amazon Personalize Monitor fournit un CloudWatch tableau de bord, des métriques personnalisées, des alarmes d'utilisation et des fonctions d'optimisation des coûts pour les campagnes Amazon Personalize. Consultez l'Amazon Personalize Monitor dans le référentiel d'exemples Amazon Personalize.

Comment définir le débit de transaction maximal pour une campagne ?

Vous ne pouvez définir que le débit minimum pour une campagne. Lorsque vous créez une campagne Amazon Personalize, vous spécifiez une capacité de transaction dédiée pour créer des

recommandations en temps réel pour les utilisateurs de votre application. Si votre TPS augmente au-delà`minProvisionedTPS`, Amazon Personalize redimensionne automatiquement la capacité allouée à la hausse ou à la baisse, mais jamais en dessous de. `minProvisionedTPS` Pour de plus amples informations, veuillez consulter Nombre minimal de transactions provisionnées par seconde et auto-scaling.

Recommandations

Comment savoir si mon modèle Amazon Personalize génère des recommandations de qualité ?

Évaluez les performances de la version de votre solution à l'aide de mesures hors ligne et en ligne (voirÉvaluation d'une version de la solution Amazon Personalize à l'aide de métriques) et de tests en ligne (tels que A/B testing). For more information about A/B les tests, voirMesurer l'impact des recommandations à l'aide A/B de tests.

Comment puis-je supprimer ma tâche d'inférence par lots et pourquoi son statut est-il « actif » ?

Vous ne pouvez pas supprimer les tâches d'inférence par lots. Lorsque le statut d'une tâche d'inférence par lots est actif, la tâche est terminée. Vous pouvez accéder à vos recommandations dans le compartiment ou le dossier de sortie Amazon S3. Le travail d'inférence par lots ne vous coûtera aucun coût supplémentaire une fois le travail terminé. Cependant, d'autres services tels qu'Amazon S3 peuvent vous facturer des frais supplémentaires pour le stockage des données d'entrée et de sortie.

Pourquoi ma campagne soutenue par des cartes SIM recommande-t-elle des éléments qui ne sont pas similaires en fonction des métadonnées ?

SIMS utilise votre jeu de données d'interactions entre les articles pour déterminer la similitude, et non les métadonnées des articles telles que la couleur ou le prix. SIMS identifie la cooccurrence de l'élément dans l'historique des utilisateurs de votre jeu de données Interaction afin de recommander des éléments similaires. Pour de plus amples informations, veuillez consulter Recette SIMS.

Puis-je obtenir plus de 500 éléments en une seule opération GetRecommendations d'API ?

500 est le nombre maximum d'objets que vous pouvez récupérer en une seule foisGetRecommendations. Cette valeur ne peut pas être augmentée.

Recommandations de filtrage

Pourquoi mes recommandations ne sont-elles pas filtrées comme prévu ?

Cela peut se produire pour diverses raisons :

- Le format ou la syntaxe de votre expression de filtre présentent peut-être un problème. Pour des exemples d'expressions de filtre correctement formatées, consultezExemples d'expressions de filtre.
- Amazon Personalize prend en compte jusqu'à 100 des interactions les plus récentes par utilisateur et par type d'événement. Il s'agit d'un quota ajustable. Vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Si vous n'importez pas les interactions entre les articles d'un utilisateur pendant trois mois, vos filtres ne tiennent plus compte des données historiques de l'utilisateur. Pour prendre en compte ces données, vous devez réimporter l'historique complet des événements de l'utilisateur.

Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Comment puis-je supprimer des recommandations des articles déjà achetés ?

Pour les groupes de données du domaine ECOMMERCE, si vous créez un recommandateur avec le Recommandé pour vous ou les cas d'Les clients ayant consulté X ont également consultéutilisation, Amazon Personalize filtre automatiquement les articles achetés par l'utilisateur en fonction de l'userID que vous spécifiez `Purchase` et des événements.

Pour les autres cas d'utilisation de groupes de données de domaines ou de ressources personnalisées, utilisez un filtre pour supprimer les articles achetés. Ajoutez un attribut de type d'`Purchase`dévénement à vos données, enregistrez les événements d'achat avec l'`PutItems`opération et créez un filtre qui supprime les articles achetés des recommandations. Par exemple :

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("purchased")
```

Pour de plus amples informations, veuillez consulter Recommandations de filtrage et segments d'utilisateurs.

Messages d'erreur courants dans Amazon Personalize

Les sections suivantes répertorient et expliquent certains des messages que vous pouvez rencontrer lors de l'utilisation d'Amazon Personalize.

Rubriques

- Importation et gestion des données
- Création d'une solution et d'une version de solution (ressources personnalisées)
- Déploiement de modèles (campagnes personnalisées)
- Recommandeurs (groupes de jeux de données de domaine)
- Recommandations
- Recommandations de filtrage

Importation et gestion des données

Message d'erreur : Emplacement des données non valide.

Assurez-vous d'avoir utilisé la bonne syntaxe pour l'emplacement de votre compartiment Amazon S3. Pour les tâches d'importation de jeux de données, utilisez la syntaxe suivante pour l'emplacement de vos données dans Amazon S3 :

`s3://amzn-s3-demo-bucket/<folder path>/<CSVfilename>`

Si vos fichiers CSV se trouvent dans un dossier et que vous souhaitez charger plusieurs fichiers avec une seule tâche d'importation de jeux de données, utilisez cette syntaxe sans le nom du fichier CSV.

Message d'erreur : Une erreur s'est produite (LimitExceededException) lors de l'appel de l' CreateDatasetImportJob opération : Plus de 5 ressources ayant le statut PENDING ou IN_PROGRESS.

Vous pouvez avoir un total de 5 tâches d'importation de jeux de données en attente ou en cours par région. Il ne s'agit pas d'un quota ajustable. Pour obtenir la liste complète des quotas pour Amazon Personalize, consultezAmazon Personalize : points de terminaison et quotas.

Message d'erreur : Impossible de créer une tâche d'importation de données pour l'ensemble de données... Privilèges insuffisants pour accéder aux données dans Amazon S3. <dataset type>

Donnez à Amazon Personalize l'accès à vos ressources Amazon S3 en associant des politiques d'accès à votre compartiment Amazon S3 et à votre rôle de service Amazon Personalize. Consultez Permettre à Amazon Personalize d'accéder aux ressources Amazon S3.

Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement, vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour de plus amples informations, veuillez consulter Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Message d'erreur : Impossible de créer un ensemble de données de tâches d'importation de données... Les colonnes suivantes sont absentes du fichier CSV en entrée : [COLUMN_NAME, COLUMN_NAME<dataset type>].

Les données que vous importez dans Amazon Personalize, y compris les noms d'attributs et les types de données, doivent correspondre au schéma du jeu de données de destination. Pour de plus amples informations, veuillez consulter Création de fichiers JSON pour les schémas Amazon Personalize.

Message d'erreur : La longueur ne peut pas dépasser le nombre de caractères pour. <character limit><COLLUMN_NAME> Si aucune valeur ne dépasse la limite de caractères, assurez-vous que vos données respectent les directives de mise en forme répertoriées dans le https://docs.aws.amazon.com/personalize/ latest/dg/data fichier -prep-formatting.html.

Vérifiez que toutes les valeurs de cette colonne ne dépassent pas la limite de caractères. Si aucune valeur ne dépasse la limite de caractères, vérifiez les éléments suivants dans les champs textuels précédents :

- Assurez-vous que toutes les données textuelles sont placées entre guillemets. Utilisez le \ caractère pour éviter les guillemets ou \ les caractères de vos données.
- Assurez-vous que chaque enregistrement de votre fichier CSV se trouve sur une seule ligne.

Création d'une solution et d'une version de solution (ressources personnalisées)

Message d'erreur : échec de la création. L'ensemble de données compte moins de 25 utilisateurs avec au moins 2 interactions chacun.

Vous devez importer davantage de données avant de pouvoir entraîner le modèle. Le nombre minimal de données requis pour former un modèle sont les suivantes :

- Au moins 1 000 enregistrements d'interactions entre articles provenant d'utilisateurs interagissant avec des articles de votre catalogue. Ces interactions peuvent provenir d'importations groupées, d'événements diffusés en continu, ou des deux.
- Au moins 25 utilisateurs uniques IDs avec au moins deux interactions avec des objets pour chacun.

Pour obtenir des recommandations en temps réel, importez davantage de données à l'aide d'une tâche d'importation de jeux de données ou enregistrez davantage d'événements d'interaction pour vos utilisateurs à l'aide d'un outil de suivi des événements et de l'PutEventsopération. Pour plus d'informations sur l'enregistrement d'événements en temps réel, consultezEnregistrement des événements en temps réel pour influencer les recommandations.

Pour les recommandations par lots, importez vos données à l'aide d'une tâche d'importation de jeux de données lorsque vous disposez de plus de données. Pour plus d'informations, sur l'importation de données en masse, voirImportation de données de formation dans les ensembles de données Amazon Personalize.

Déploiement de modèles (campagnes personnalisées)

Erreur : Impossible de créer une campagne. Plus de 5 ressources à l'état ACTIF. Supprimez-en certaines et réessayez.

Vous pouvez avoir un total de 5 campagnes Amazon Personalize actives par groupe de jeux de données. Ce quota est ajustable et vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Pour obtenir la liste complète des limites et des quotas pour Amazon Personalize, consultezAmazon Personalize : points de terminaison et quotas.

Recommandeurs (groupes de jeux de données de domaine)

Erreur : l'ensemble de données comporte moins de 1 000 interactions après le filtrage par type d'événement : <event type>

Les différents cas d'utilisation nécessitent différents types d'événements. Vos données doivent contenir au moins 1 000 événements du type requis pour votre cas d'utilisation. Pour de plus amples informations, consultez Choix d'un cas d'utilisation.

Recommandations

Message d'erreur de la tâche d'inférence par lots : chemin d'entrée S3 non valide ou chemin de sortie S3 non valide

Assurez-vous d'utiliser la syntaxe correcte pour vos emplacements d'entrée ou de sortie Amazon S3. Assurez-vous également que votre emplacement de sortie est différent de vos données d'entrée. Il doit s'agir d'un dossier dans le même compartiment Amazon S3 ou dans un autre compartiment.

Utilisez la syntaxe suivante pour l'emplacement du fichier d'entrée dans Amazon S3 : **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>`**

Utilisez la syntaxe suivante pour le dossier de sortie dans Amazon S3 : **`s3://amzn-s3-demo-bucket/<output folder name>/`**

Recommandations de filtrage

Message d'erreur : Impossible de créer le filtre. Symbole de saisie non valide : $parameterName. Les espaces réservés ne sont pas autorisés avec l'opérateur NOT_IN.

Vous ne pouvez pas utiliser de paramètres d'espace réservé dans une expression de filtre qui utilise l'opérateur NOT_IN. Utilisez plutôt l'opérateur IN et utilisez l'action opposée : utilisez Include au lieu de Exclude (ou l'inverse).

Par exemple, si vous souhaitez utiliser`INCLUDE ItemID WHERE Items.GENRE NOT IN ($GENRE)`, vous pouvez utiliser `EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE)` et obtenir les mêmes résultats.

Pour plus d’informations sur les filtres, consultez Filtrer les éléments d'expression.

Message d'erreur : Impossible de créer le filtre. Expression non valide... lors du filtrage sur des champs de type booléen

Vous ne pouvez pas créer d'expressions de filtre qui filtrent à l'aide de valeurs de type booléen dans votre schéma. Pour filtrer en fonction de valeurs booléennes, utilisez un schéma avec un champ de type `String` et utilisez les valeurs `True` et `False` dans vos données. Vous pouvez également utiliser le type `int` ou les valeurs `long` `0` et`1`.

Pour plus d’informations sur les filtres, consultez Filtrer les éléments d'expression.

Spécifier les ressources avec AWS CloudFormation

Amazon Personalize est intégré à AWS CloudFormation un service qui vous aide à modéliser et à configurer vos AWS ressources afin que vous puissiez passer moins de temps à créer et à gérer vos ressources et votre infrastructure. Vous créez un modèle qui décrit toutes les AWS ressources que vous pouvez spécifier (telles que les groupes de jeux de données Amazon Personalize). CloudFormation fournit et configure ensuite ces ressources pour vous.

Lorsque vous l'utilisez CloudFormation, vous pouvez réutiliser votre modèle pour configurer vos ressources Amazon Personalize de manière cohérente et répétée. Décrivez vos ressources une seule fois, puis fournissez les mêmes ressources encore et encore dans plusieurs Comptes AWS régions.

Rubriques

- Amazon Personalize et CloudFormation modèles
- Exemples CloudFormation de modèles pour les ressources Amazon Personalize
- En savoir plus sur CloudFormation

Amazon Personalize et CloudFormation modèles

Pour fournir et configurer des ressources pour Amazon Personalize et les services associés, vous devez comprendre les CloudFormation modèles. Les modèles sont des fichiers texte formatés en JSON ou YAML. Ces modèles décrivent les ressources que vous souhaitez mettre à disposition dans vos CloudFormation piles. Si vous n'êtes pas familiarisé avec JSON ou YAML, vous pouvez utiliser CloudFormation Designer pour vous aider à démarrer avec les CloudFormation modèles. Pour plus d'informations, consultez Qu'est-ce que CloudFormation Designer dans le Guide de l'utilisateur AWS CloudFormation .

Amazon Personalize prend en charge la spécification d'ensembles de données, de groupes de données, de tâches d'importation de jeux de données, de schémas et de solutions dans. CloudFormation Pour plus d'informations, consultez la référence au type de ressource Amazon Personalize dans le guide de AWS CloudFormation l'utilisateur.

Exemples CloudFormation de modèles pour les ressources Amazon Personalize

Les exemples de CloudFormation modèles suivants vous montrent comment spécifier différentes ressources Amazon Personalize.

Rubriques

- CreateDatasetGroup
- CreateDataset
- CreateSchema
- CreateSolution

CreateDatasetGroup

JSON

```
{
   "AWSTemplateFormatVersion":"2010-09-09",
   "Resources":{
      "MyDatasetGroup": {
            "Type": "AWS::Personalize::DatasetGroup",
            "Properties": {
               "Name": "my-dataset-group-name"
            }
      }
   }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDatasetGroup:
    Type: 'AWS::Personalize::DatasetGroup'
    Properties:
      Name: my-dataset-group-name
```

CreateDataset

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyDataset": {
      "Type": "AWS::Personalize::Dataset",
      "Properties": {
        "Name": "my-dataset-name",
        "DatasetType": "Interactions",
        "DatasetGroupArn": "arn:aws:personalize:us-west-2:123456789012:dataset-
group/dataset-group-name",
        "SchemaArn": "arn:aws:personalize:us-west-2:123456789012:schema/schema-
name",
        "DatasetImportJob": {
          "JobName": "my-import-job-name",
          "DataSource": {
            "DataLocation": "s3://amzn-s3-demo-bucket/file-name.csv"
          },
          "RoleArn": "arn:aws:iam::123456789012:role/personalize-role"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDataset:
    Type: 'AWS::Personalize::Dataset'
    Properties:
      Name: my-dataset-name
      DatasetType: Interactions
      DatasetGroupArn: 'arn:aws:personalize:us-west-2:123456789012:dataset-group/
dataset-group-name'
      SchemaArn: 'arn:aws:personalize:us-west-2:123456789012:schema/schema-name'
      DatasetImportJob:
        JobName: my-import-job-name
        DataSource:
```

```
          DataLocation: 's3://amzn-s3-demo-bucket/file-name.csv'
        RoleArn: 'arn:aws:iam::123456789012:role/personalize-role'
```

CreateSchema

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "MySchema": {
            "Type": "AWS::Personalize::Schema",
            "Properties": {
                "Name": "my-schema-name",
                "Schema": "{\"type\": \"record\",\"name\": \"Interactions\",
 \"namespace\": \"com.amazonaws.personalize.schema\", \"fields\": [ { \"name\":
 \"USER_ID\", \"type\": \"string\" }, { \"name\": \"ITEM_ID\", \"type\": \"string
\" }, { \"name\": \"TIMESTAMP\", \"type\": \"long\"}], \"version\": \"1.0\"}"
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySchema:
    Type: AWS::Personalize::Schema
    Properties:
      Name: "my-schema-name"
      Schema: >-
        {"type": "record","name": "Interactions", "namespace":
        "com.amazonaws.personalize.schema", "fields": [ { "name": "USER_ID",
        "type": "string" }, { "name": "ITEM_ID", "type": "string" }, { "name":
        "TIMESTAMP", "type": "long"}], "version": "1.0"}
```

CreateSolution

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "MySolution": {
            "Type": "AWS::Personalize::Solution",
            "Properties": {
               "Name": "my-solution-name",
               "DatasetGroupArn": "arn:aws:personalize:us-
west-2:123456789012:dataset-group/my-dataset-group-name",
               "RecipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
               "SolutionConfig": {
                  "EventValueThreshold" : ".05"
                }
            }
         }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySolution:
    Type: 'AWS::Personalize::Solution'
    Properties:
      Name: my-solution-name
      DatasetGroupArn: >-
        arn:aws:personalize:us-west-2:123456789012:dataset-group/my-dataset-group-
name
      RecipeArn: 'arn:aws:personalize:::recipe/aws-user-personalization'
      SolutionConfig:
        EventValueThreshold: '.05'
```

En savoir plus sur CloudFormation

Pour en savoir plus CloudFormation, consultez les ressources suivantes :

- AWS CloudFormation
- AWS CloudFormation guide de l'utilisateur
- CloudFormation Référence d’API
- AWS CloudFormation Guide de l'utilisateur de l'interface de ligne de commande

Exemples de code pour Amazon Personalize à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Personalize avec un kit de développement AWS logiciel (SDK).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de code pour Amazon Personalize à l'aide de AWS SDKs
  - Exemples de base pour l'utilisation d'Amazon Personalize AWS SDKs
    - Actions utilisées par Amazon Personalize AWS SDKs
      - Utilisation CreateBatchInferenceJob avec un AWS SDK
      - Utilisation CreateBatchSegmentJob avec un AWS SDK
      - Utilisation CreateCampaign avec un AWS SDK
      - Utilisation CreateDataset avec un AWS SDK
      - Utilisation CreateDatasetExportJob avec un AWS SDK
      - Utilisation CreateDatasetGroup avec un AWS SDK
      - Utilisation CreateDatasetImportJob avec un AWS SDK
      - Utilisation CreateEventTracker avec un AWS SDK
      - Utilisation CreateFilter avec un AWS SDK
      - Utilisation CreateRecommender avec un AWS SDK
      - Utilisation CreateSchema avec un AWS SDK
      - Utilisation CreateSolution avec un AWS SDK
      - Utilisation CreateSolutionVersion avec un AWS SDK
      - Utilisation DeleteCampaign avec un AWS SDK
      - Utilisation DeleteEventTracker avec un AWS SDK
      - Utilisation DeleteSolution avec un AWS SDK
      - Utilisation DescribeCampaign avec un AWS SDK

        - Utilisation DescribeRecipe avec un AWS SDK
        - Utilisation DescribeSolution avec un AWS SDK
        - Utilisation ListCampaigns avec un AWS SDK
        - Utilisation ListDatasetGroups avec un AWS SDK
        - Utilisation ListRecipes avec un AWS SDK
        - Utilisation ListSolutions avec un AWS SDK
        - Utilisation UpdateCampaign avec un AWS SDK
- Exemples de code pour Amazon Personalize Events à l'aide de AWS SDKs
  - Exemples de base pour Amazon Personalize Events utilisant AWS SDKs
    - Actions pour Amazon Personalize Events à l'aide de AWS SDKs
      - Utilisation PutEvents avec un AWS SDK
      - Utilisation PutItems avec un AWS SDK
      - Utilisation PutUsers avec un AWS SDK
- Exemples de code pour Amazon Personalize Runtime à l'aide de AWS SDKs
  - Exemples de base pour Amazon Personalize Runtime utilisant AWS SDKs
    - Actions pour Amazon Personalize Runtime à l'aide de AWS SDKs
      - Utilisation GetPersonalizedRanking avec un AWS SDK
      - Utilisation GetRecommendations avec un AWS SDK

Exemples de code pour Amazon Personalize à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Personalize avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Amazon Personalize

- Exemples de base pour l'utilisation d'Amazon Personalize AWS SDKs
  - Actions utilisées par Amazon Personalize AWS SDKs
    - Utilisation CreateBatchInferenceJob avec un AWS SDK
    - Utilisation CreateBatchSegmentJob avec un AWS SDK
    - Utilisation CreateCampaign avec un AWS SDK
    - Utilisation CreateDataset avec un AWS SDK
    - Utilisation CreateDatasetExportJob avec un AWS SDK
    - Utilisation CreateDatasetGroup avec un AWS SDK
    - Utilisation CreateDatasetImportJob avec un AWS SDK
    - Utilisation CreateEventTracker avec un AWS SDK
    - Utilisation CreateFilter avec un AWS SDK
    - Utilisation CreateRecommender avec un AWS SDK
    - Utilisation CreateSchema avec un AWS SDK
    - Utilisation CreateSolution avec un AWS SDK
    - Utilisation CreateSolutionVersion avec un AWS SDK
    - Utilisation DeleteCampaign avec un AWS SDK
    - Utilisation DeleteEventTracker avec un AWS SDK
    - Utilisation DeleteSolution avec un AWS SDK
    - Utilisation DescribeCampaign avec un AWS SDK
    - Utilisation DescribeRecipe avec un AWS SDK
    - Utilisation DescribeSolution avec un AWS SDK
    - Utilisation ListCampaigns avec un AWS SDK
    - Utilisation ListDatasetGroups avec un AWS SDK
    - Utilisation ListRecipes avec un AWS SDK
    - Utilisation ListSolutions avec un AWS SDK
    - Utilisation UpdateCampaign avec un AWS SDK

Exemples de base pour l'utilisation d'Amazon Personalize AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Amazon Personalize with AWS SDKs.

Exemples

- Actions utilisées par Amazon Personalize AWS SDKs
  - Utilisation CreateBatchInferenceJob avec un AWS SDK
  - Utilisation CreateBatchSegmentJob avec un AWS SDK
  - Utilisation CreateCampaign avec un AWS SDK
  - Utilisation CreateDataset avec un AWS SDK
  - Utilisation CreateDatasetExportJob avec un AWS SDK
  - Utilisation CreateDatasetGroup avec un AWS SDK
  - Utilisation CreateDatasetImportJob avec un AWS SDK
  - Utilisation CreateEventTracker avec un AWS SDK
  - Utilisation CreateFilter avec un AWS SDK
  - Utilisation CreateRecommender avec un AWS SDK
  - Utilisation CreateSchema avec un AWS SDK
  - Utilisation CreateSolution avec un AWS SDK
  - Utilisation CreateSolutionVersion avec un AWS SDK
  - Utilisation DeleteCampaign avec un AWS SDK
  - Utilisation DeleteEventTracker avec un AWS SDK
  - Utilisation DeleteSolution avec un AWS SDK
  - Utilisation DescribeCampaign avec un AWS SDK
  - Utilisation DescribeRecipe avec un AWS SDK
  - Utilisation DescribeSolution avec un AWS SDK
  - Utilisation ListCampaigns avec un AWS SDK
  - Utilisation ListDatasetGroups avec un AWS SDK
  - Utilisation ListRecipes avec un AWS SDK
  - Utilisation ListSolutions avec un AWS SDK
  - Utilisation UpdateCampaign avec un AWS SDK

Actions utilisées par Amazon Personalize AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Amazon Personalize individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez la Référence des API Amazon Personalize.

Exemples

- Utilisation CreateBatchInferenceJob avec un AWS SDK
- Utilisation CreateBatchSegmentJob avec un AWS SDK
- Utilisation CreateCampaign avec un AWS SDK
- Utilisation CreateDataset avec un AWS SDK
- Utilisation CreateDatasetExportJob avec un AWS SDK
- Utilisation CreateDatasetGroup avec un AWS SDK
- Utilisation CreateDatasetImportJob avec un AWS SDK
- Utilisation CreateEventTracker avec un AWS SDK
- Utilisation CreateFilter avec un AWS SDK
- Utilisation CreateRecommender avec un AWS SDK
- Utilisation CreateSchema avec un AWS SDK
- Utilisation CreateSolution avec un AWS SDK
- Utilisation CreateSolutionVersion avec un AWS SDK
- Utilisation DeleteCampaign avec un AWS SDK
- Utilisation DeleteEventTracker avec un AWS SDK
- Utilisation DeleteSolution avec un AWS SDK
- Utilisation DescribeCampaign avec un AWS SDK
- Utilisation DescribeRecipe avec un AWS SDK
- Utilisation DescribeSolution avec un AWS SDK
- Utilisation ListCampaigns avec un AWS SDK
- Utilisation ListDatasetGroups avec un AWS SDK
- Utilisation ListRecipes avec un AWS SDK

- Utilisation ListSolutions avec un AWS SDK
- Utilisation UpdateCampaign avec un AWS SDK

Utilisation **`CreateBatchInferenceJob`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateBatchInferenceJob`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        public static String createPersonalizeBatchInferenceJob(PersonalizeClient
personalizeClient,
                        String solutionVersionArn,
                        String jobName,
                        String s3InputDataSourcePath,
                        String s3DataDestinationPath,
                        String roleArn,
                        String explorationWeight,
                        String explorationItemAgeCutOff) {

                long waitInMilliseconds = 60 * 1000;
                String status;
                String batchInferenceJobArn;

                try {

                        // Set up data input and output parameters.
                        S3DataConfig inputSource = S3DataConfig.builder()
                                        .path(s3InputDataSourcePath)
                                        .build();

                        S3DataConfig outputDestination = S3DataConfig.builder()
                                        .path(s3DataDestinationPath)
                                        .build();
```

```
                        BatchInferenceJobInput jobInput =
BatchInferenceJobInput.builder()
                                        .s3DataSource(inputSource)
                                        .build();

                        BatchInferenceJobOutput jobOutputLocation =
BatchInferenceJobOutput.builder()
                                        .s3DataDestination(outputDestination)
                                        .build();

                        // Optional code to build the User-Personalization
specific item exploration
                        // config.
                        HashMap<String, String> explorationConfig = new
HashMap<>();

                        explorationConfig.put("explorationWeight",
explorationWeight);
                        explorationConfig.put("explorationItemAgeCutOff",
explorationItemAgeCutOff);

                        BatchInferenceJobConfig jobConfig =
BatchInferenceJobConfig.builder()
                                        .itemExplorationConfig(explorationConfig)
                                        .build();

                        // End optional User-Personalization recipe specific
code.

                        CreateBatchInferenceJobRequest
createBatchInferenceJobRequest = CreateBatchInferenceJobRequest
                                        .builder()
                                        .solutionVersionArn(solutionVersionArn)
                                        .jobInput(jobInput)
                                        .jobOutput(jobOutputLocation)
                                        .jobName(jobName)
                                        .roleArn(roleArn)
                                        .batchInferenceJobConfig(jobConfig) //
Optional
                                        .build();

                        batchInferenceJobArn =
personalizeClient.createBatchInferenceJob(createBatchInferenceJobRequest)
```

```
                                        .batchInferenceJobArn();

                        DescribeBatchInferenceJobRequest
describeBatchInferenceJobRequest = DescribeBatchInferenceJobRequest
                                        .builder()

.batchInferenceJobArn(batchInferenceJobArn)
                                        .build();

                        long maxTime = Instant.now().getEpochSecond() + 3 * 60 *
60;
                        while (Instant.now().getEpochSecond() < maxTime) {

                                BatchInferenceJob batchInferenceJob =
personalizeClient

.describeBatchInferenceJob(describeBatchInferenceJobRequest)
                                                .batchInferenceJob();

                                status = batchInferenceJob.status();
                                System.out.println("Batch inference job status: "
+ status);

                                if (status.equals("ACTIVE") ||
status.equals("CREATE FAILED")) {
                                        break;
                                }
                                try {
                                        Thread.sleep(waitInMilliseconds);
                                } catch (InterruptedException e) {
                                        System.out.println(e.getMessage());
                                }
                        }
                        return batchInferenceJobArn;

                } catch (PersonalizeException e) {
                        System.out.println(e.awsErrorDetails().errorMessage());
                }
                return "";
        }
```

- Pour plus de détails sur l'API, reportez-vous CreateBatchInferenceJobà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchInferenceJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch inference job's parameters.

export const createBatchInferenceJobParam = {
  jobName: "JOB_NAME",
  jobInput: {
    s3DataSource: {
      path: "INPUT_PATH",
    },
  },
  jobOutput: {
    s3DataDestination: {
      path: "OUTPUT_PATH",
    },
  },
  roleArn: "ROLE_ARN",
  solutionVersionArn: "SOLUTION_VERSION_ARN",
  numResults: 20,
};

export const run = async () => {
  try {
```

```
    const response = await personalizeClient.send(
      new CreateBatchInferenceJobCommand(createBatchInferenceJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateBatchInferenceJobà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateBatchSegmentJob`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateBatchSegmentJob`.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchSegmentJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});
```

```
// Set the batch segment job's parameters.

export const createBatchSegmentJobParam = {
  jobName: "NAME",
  jobInput: {
    s3DataSource: {
      path: "INPUT_PATH",
    },
  },
  jobOutput: {
    s3DataDestination: {
      path: "OUTPUT_PATH",
    },
  },
  roleArn: "ROLE_ARN",
  solutionVersionArn: "SOLUTION_VERSION_ARN",
  numResults: 20,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateBatchSegmentJobCommand(createBatchSegmentJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateBatchSegmentJobà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateCampaign`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateCampaign`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createPersonalCompaign(PersonalizeClient
personalizeClient, String solutionVersionArn,
           String name) {

       try {
           CreateCampaignRequest createCampaignRequest =
CreateCampaignRequest.builder()
                   .minProvisionedTPS(1)
                   .solutionVersionArn(solutionVersionArn)
                   .name(name)
                   .build();

           CreateCampaignResponse campaignResponse =
personalizeClient.createCampaign(createCampaignRequest);
           System.out.println("The campaign ARN is " +
campaignResponse.campaignArn());
           return campaignResponse.campaignArn();
       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return null;
   }
```

- Pour plus de détails sur l'API, reportez-vous CreateCampaignà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.

import { CreateCampaignCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the campaign's parameters.
export const createCampaignParam = {
  solutionVersionArn: "SOLUTION_VERSION_ARN" /* required */,
  name: "NAME" /* required */,
  minProvisionedTPS: 1 /* optional integer */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateCampaignCommand(createCampaignParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateCampaignà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateDataset`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateDataset`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
public static String createDataset(PersonalizeClient personalizeClient,
        String datasetName,
        String datasetGroupArn,
        String datasetType,
        String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
                .name(datasetName)
                .datasetGroupArn(datasetGroupArn)
                .datasetType(datasetType)
                .schemaArn(schemaArn)
                .build();

        String datasetArn = personalizeClient.createDataset(request)
                .datasetArn();
        System.out.println("Dataset " + datasetName + " created.");
        return datasetArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  datasetType: "DATASET_TYPE" /* required */,
  name: "NAME" /* required */,
  schemaArn: "SCHEMA_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetCommand(createDatasetParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
```

```
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateDatasetExportJob`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateDatasetExportJob`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createDatasetExportJob(PersonalizeClient
personalizeClient,
           String jobName,
           String datasetArn,
           IngestionMode ingestionMode,
           String roleArn,
           String s3BucketPath,
           String kmsKeyArn) {

       long waitInMilliseconds = 30 * 1000; // 30 seconds
       String status = null;

       try {

           S3DataConfig exportS3DataConfig =
S3DataConfig.builder().path(s3BucketPath).kmsKeyArn(kmsKeyArn).build();
```

```
            DatasetExportJobOutput jobOutput =
DatasetExportJobOutput.builder().s3DataDestination(exportS3DataConfig)
                    .build();

            CreateDatasetExportJobRequest createRequest =
CreateDatasetExportJobRequest.builder()
                    .jobName(jobName)
                    .datasetArn(datasetArn)
                    .ingestionMode(ingestionMode)
                    .jobOutput(jobOutput)
                    .roleArn(roleArn)
                    .build();

            String datasetExportJobArn =
personalizeClient.createDatasetExportJob(createRequest).datasetExportJobArn();

            DescribeDatasetExportJobRequest describeDatasetExportJobRequest =
DescribeDatasetExportJobRequest.builder()
                    .datasetExportJobArn(datasetExportJobArn)
                    .build();

            long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                DatasetExportJob datasetExportJob = personalizeClient

.describeDatasetExportJob(describeDatasetExportJobRequest)
                        .datasetExportJob();

                status = datasetExportJob.status();
                System.out.println("Export job status: " + status);

                if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                    return status;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
```

```
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetExportJobà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetExportJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the export job parameters.
export const datasetExportJobParam = {
  datasetArn: "DATASET_ARN" /* required */,
  jobOutput: {
    s3DataDestination: {
      path: "S3_DESTINATION_PATH" /* required */,
      //kmsKeyArn: 'ARN'  /* include if your bucket uses AWS KMS for encryption
    },
  },
  jobName: "NAME" /* required */,
  roleArn: "ROLE_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
```

```
      new CreateDatasetExportJobCommand(datasetExportJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetExportJobà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateDatasetGroup`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateDatasetGroup`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createDatasetGroup(PersonalizeClient personalizeClient,
String datasetGroupName) {

       try {
           CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
                   .name(datasetGroupName)
                   .build();
```

```
            return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

Créez un groupe de jeu de données de domaine.

```
    public static String createDomainDatasetGroup(PersonalizeClient
personalizeClient,
            String datasetGroupName,
            String domain) {

        try {
            CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
                    .name(datasetGroupName)
                    .domain(domain)
                    .build();
            return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetGroupà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.

import { CreateDatasetGroupCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset group parameters.
export const createDatasetGroupParam = {
  name: "NAME" /* required */,
};

export const run = async (createDatasetGroupParam) => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetGroupCommand(createDatasetGroupParam),
    );
    console.log("Success", response);
    return "Run successfully"; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run(createDatasetGroupParam);
```

Créez un groupe de jeu de données de domaine.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
```

```
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: "NAME" /* required */,
  domain:
    "DOMAIN" /* required for a domain dsg, specify ECOMMERCE or VIDEO_ON_DEMAND
 */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetGroupCommand(domainDatasetGroupParams),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetGroupà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateDatasetImportJob`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateDatasetImportJob`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
           String jobName,
           String datasetArn,
           String s3BucketPath,
           String roleArn) {

       long waitInMilliseconds = 60 * 1000;
       String status;
       String datasetImportJobArn;

       try {
           DataSource importDataSource = DataSource.builder()
                   .dataLocation(s3BucketPath)
                   .build();

           CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
                   .datasetArn(datasetArn)
                   .dataSource(importDataSource)
                   .jobName(jobName)
                   .roleArn(roleArn)
                   .build();

           datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
                   .datasetImportJobArn();
           DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
                   .datasetImportJobArn(datasetImportJobArn)
                   .build();
```

```
            long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                DatasetImportJob datasetImportJob = personalizeClient

.describeDatasetImportJob(describeDatasetImportJobRequest)
                        .datasetImportJob();

                status = datasetImportJob.status();
                System.out.println("Dataset import job status: " + status);

                if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return datasetImportJobArn;

        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetImportJobà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetImportJobCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: "DATASET_ARN" /* required */,
  dataSource: {
    /* required */
    dataLocation: "S3_PATH",
  },
  jobName: "NAME" /* required */,
  roleArn: "ROLE_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateDatasetImportJobCommand(datasetImportJobParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateDatasetImportJobà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateEventTracker`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateEventTracker`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createEventTracker(PersonalizeClient personalizeClient,
String eventTrackerName,
           String datasetGroupArn) {

       String eventTrackerId = "";
       String eventTrackerArn;
       long maxTime = 3 * 60 * 60; // 3 hours
       long waitInMilliseconds = 20 * 1000; // 20 seconds
       String status;

       try {

           CreateEventTrackerRequest createEventTrackerRequest =
CreateEventTrackerRequest.builder()
                   .name(eventTrackerName)
                   .datasetGroupArn(datasetGroupArn)
                   .build();

           CreateEventTrackerResponse createEventTrackerResponse =
personalizeClient
                   .createEventTracker(createEventTrackerRequest);

           eventTrackerArn = createEventTrackerResponse.eventTrackerArn();
           eventTrackerId = createEventTrackerResponse.trackingId();
           System.out.println("Event tracker ARN: " + eventTrackerArn);
           System.out.println("Event tracker ID: " + eventTrackerId);

           maxTime = Instant.now().getEpochSecond() + maxTime;

           DescribeEventTrackerRequest describeRequest =
DescribeEventTrackerRequest.builder()
                   .eventTrackerArn(eventTrackerArn)
```

```
                    .build();

            while (Instant.now().getEpochSecond() < maxTime) {

                status =
personalizeClient.describeEventTracker(describeRequest).eventTracker().status();
                System.out.println("EventTracker status: " + status);

                if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return eventTrackerId;
        } catch (PersonalizeException e) {
            System.out.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return eventTrackerId;
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateEventTrackerà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateEventTrackerCommand } from "@aws-sdk/client-personalize";
```

```
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the event tracker's parameters.
export const createEventTrackerParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  name: "NAME" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateEventTrackerCommand(createEventTrackerParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateEventTrackerà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateFilter`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateFilter`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
public static String createFilter(PersonalizeClient personalizeClient,
        String filterName,
        String datasetGroupArn,
        String filterExpression) {
    try {
        CreateFilterRequest request = CreateFilterRequest.builder()
                .name(filterName)
                .datasetGroupArn(datasetGroupArn)
                .filterExpression(filterExpression)
                .build();

        return personalizeClient.createFilter(request).filterArn();
    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFilterà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateFilterCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the filter's parameters.
export const createFilterParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  name: "NAME" /* required */,
  filterExpression: "FILTER_EXPRESSION" /*required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateFilterCommand(createFilterParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateFilterà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateRecommender`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateRecommender`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public static String createRecommender(PersonalizeClient personalizeClient,
            String name,
            String datasetGroupArn,
            String recipeArn) {

        long maxTime = 0;
        long waitInMilliseconds = 30 * 1000; // 30 seconds
        String recommenderStatus = "";

        try {
            CreateRecommenderRequest createRecommenderRequest =
CreateRecommenderRequest.builder()
                    .datasetGroupArn(datasetGroupArn)
                    .name(name)
                    .recipeArn(recipeArn)
                    .build();

            CreateRecommenderResponse recommenderResponse = personalizeClient
                    .createRecommender(createRecommenderRequest);
            String recommenderArn = recommenderResponse.recommenderArn();
            System.out.println("The recommender ARN is " + recommenderArn);

            DescribeRecommenderRequest describeRecommenderRequest =
DescribeRecommenderRequest.builder()
```

```
                    .recommenderArn(recommenderArn)
                    .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            while (Instant.now().getEpochSecond() < maxTime) {

                recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                        .status();
                System.out.println("Recommender status: " + recommenderStatus);

                if (recommenderStatus.equals("ACTIVE") ||
recommenderStatus.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
            return recommenderArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateRecommenderà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateRecommenderCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the recommender's parameters.
export const createRecommenderParam = {
  name: "NAME" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateRecommenderCommand(createRecommenderParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateRecommenderà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateSchema`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateSchema`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createSchema(PersonalizeClient personalizeClient, String
schemaName, String filePath) {

       String schema = null;
       try {
           schema = new String(Files.readAllBytes(Paths.get(filePath)));
       } catch (IOException e) {
           System.out.println(e.getMessage());
       }

       try {
           CreateSchemaRequest createSchemaRequest =
CreateSchemaRequest.builder()
                   .name(schemaName)
                   .schema(schema)
                   .build();

           String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

           System.out.println("Schema arn: " + schemaArn);

           return schemaArn;
```

```
        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

Créez un schéma avec un domaine.

```
    public static String createDomainSchema(PersonalizeClient personalizeClient,
String schemaName, String domain,
            String filePath) {

        String schema = null;
        try {
            schema = new String(Files.readAllBytes(Paths.get(filePath)));
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }

        try {
            CreateSchemaRequest createSchemaRequest =
CreateSchemaRequest.builder()
                    .name(schemaName)
                    .domain(domain)
                    .schema(schema)
                    .build();

            String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

            System.out.println("Schema arn: " + schemaArn);

            return schemaArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateSchemaà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from "node:fs";

const schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = "TEST"; // For unit tests.
}
// Set the schema parameters.
export const createSchemaParam = {
  name: "NAME" /* required */,
  schema: mySchema /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSchemaCommand(createSchemaParam),
    );
```

```
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Créez un schéma avec un domaine.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from "node:fs";

const schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = "TEST"; // for unit tests.
}

// Set the domain schema parameters.
export const createDomainSchemaParam = {
  name: "NAME" /* required */,
  schema: mySchema /* required */,
  domain:
    "DOMAIN" /* required for a domain dataset group, specify ECOMMERCE or
 VIDEO_ON_DEMAND */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSchemaCommand(createDomainSchemaParam),
    );
```

```
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateSchemaà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateSolution`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateSolution`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createPersonalizeSolution(PersonalizeClient
personalizeClient,
           String datasetGroupArn,
           String solutionName,
           String recipeArn) {

       try {
           CreateSolutionRequest solutionRequest =
CreateSolutionRequest.builder()
                   .name(solutionName)
                   .datasetGroupArn(datasetGroupArn)
```

```
                    .recipeArn(recipeArn)
                    .build();

            CreateSolutionResponse solutionResponse =
 personalizeClient.createSolution(solutionRequest);
            return solutionResponse.solutionArn();

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateSolutionà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSolutionCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution parameters.
export const createSolutionParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  name: "NAME" /* required */,
};
```

```
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionCommand(createSolutionParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateSolutionà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateSolutionVersion`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateSolutionVersion`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static String createPersonalizeSolutionVersion(PersonalizeClient
personalizeClient, String solutionArn) {
       long maxTime = 0;
       long waitInMilliseconds = 30 * 1000; // 30 seconds
       String solutionStatus = "";
       String solutionVersionStatus = "";
```

```
        String solutionVersionArn = "";

        try {
            DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
                    .solutionArn(solutionArn)
                    .build();

            maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

            // Wait until solution is active.
            while (Instant.now().getEpochSecond() < maxTime) {

                solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
                System.out.println("Solution status: " + solutionStatus);

                if (solutionStatus.equals("ACTIVE") ||
solutionStatus.equals("CREATE FAILED")) {
                    break;
                }
                try {
                    Thread.sleep(waitInMilliseconds);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }

            if (solutionStatus.equals("ACTIVE")) {

                CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
                        .solutionArn(solutionArn)
                        .build();

                CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient
                        .createSolutionVersion(createSolutionVersionRequest);
                solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

                System.out.println("Solution version ARN: " +
solutionVersionArn);
```

```
                DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
                        .solutionVersionArn(solutionVersionArn)
                        .build();

                while (Instant.now().getEpochSecond() < maxTime) {

                    solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
                            .solutionVersion().status();
                    System.out.println("Solution version status: " +
solutionVersionStatus);

                    if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                        break;
                    }
                    try {
                        Thread.sleep(waitInMilliseconds);
                    } catch (InterruptedException e) {
                        System.out.println(e.getMessage());
                    }
                }
                return solutionVersionArn;
            }
        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateSolutionVersionà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSolutionVersionCommand } from "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution version parameters.
export const solutionVersionParam = {
  solutionArn: "SOLUTION_ARN" /* required */,
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionVersionCommand(solutionVersionParam),
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous CreateSolutionVersionà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteCampaign`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteCampaign`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void deleteSpecificCampaign(PersonalizeClient
personalizeClient, String campaignArn) {
       try {
           DeleteCampaignRequest campaignRequest =
DeleteCampaignRequest.builder()
                   .campaignArn(campaignArn)
                   .build();

           personalizeClient.deleteCampaign(campaignRequest);
           System.out.println("Delete request sent successfully.");
       } catch (PersonalizeException e) {
           System.err.println("Error deleting campaign: " +
e.awsErrorDetails().errorMessage());
           throw new RuntimeException(e);
       }
   }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteCampaignà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteEventTracker`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteEventTracker`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void deleteEventTracker(PersonalizeClient personalizeClient,
String eventTrackerArn) {
       try {
           DeleteEventTrackerRequest deleteEventTrackerRequest =
DeleteEventTrackerRequest.builder()
                   .eventTrackerArn(eventTrackerArn)
                   .build();

           int status =
personalizeClient.deleteEventTracker(deleteEventTrackerRequest).sdkHttpResponse().status

           System.out.println("Status code:" + status);

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DeleteEventTrackerà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteSolution`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteSolution`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void deleteGivenSolution(PersonalizeClient personalizeClient,
String solutionArn) {

       try {
           DeleteSolutionRequest solutionRequest =
DeleteSolutionRequest.builder()
                   .solutionArn(solutionArn)
                   .build();

           personalizeClient.deleteSolution(solutionRequest);
           System.out.println("Done");

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DeleteSolutionà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DescribeCampaign`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DescribeCampaign`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void describeSpecificCampaign(PersonalizeClient
personalizeClient, String campaignArn) {

       try {
           DescribeCampaignRequest campaignRequest =
DescribeCampaignRequest.builder()
                   .campaignArn(campaignArn)
                   .build();

           DescribeCampaignResponse campaignResponse =
personalizeClient.describeCampaign(campaignRequest);
           Campaign myCampaign = campaignResponse.campaign();
           System.out.println("The Campaign name is " + myCampaign.name());
           System.out.println("The Campaign status is " + myCampaign.status());

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DescribeCampaignà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DescribeRecipe`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DescribeRecipe`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void describeSpecificRecipe(PersonalizeClient
personalizeClient, String recipeArn) {

       try {
           DescribeRecipeRequest recipeRequest = DescribeRecipeRequest.builder()
                   .recipeArn(recipeArn)
                   .build();

           DescribeRecipeResponse recipeResponse =
personalizeClient.describeRecipe(recipeRequest);
           System.out.println("The recipe name is " +
recipeResponse.recipe().name());

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DescribeRecipeà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DescribeSolution`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DescribeSolution`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void describeSpecificSolution(PersonalizeClient
personalizeClient, String solutionArn) {

       try {
           DescribeSolutionRequest solutionRequest =
DescribeSolutionRequest.builder()
                   .solutionArn(solutionArn)
                   .build();

           DescribeSolutionResponse response =
personalizeClient.describeSolution(solutionRequest);
           System.out.println("The Solution name is " +
response.solution().name());

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DescribeSolutionà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`ListCampaigns`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`ListCampaigns`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void listAllCampaigns(PersonalizeClient personalizeClient,
String solutionArn) {

       try {
           ListCampaignsRequest campaignsRequest =
ListCampaignsRequest.builder()
                   .maxResults(10)
                   .solutionArn(solutionArn)
                   .build();

           ListCampaignsResponse response =
personalizeClient.listCampaigns(campaignsRequest);
           List<CampaignSummary> campaigns = response.campaigns();
           for (CampaignSummary campaign : campaigns) {
               System.out.println("Campaign name is : " + campaign.name());
               System.out.println("Campaign ARN is : " +
campaign.campaignArn());
           }

       } catch (PersonalizeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous ListCampaignsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`ListDatasetGroups`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`ListDatasetGroups`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void listDSGroups(PersonalizeClient personalizeClient) {

       try {
           ListDatasetGroupsRequest groupsRequest =
ListDatasetGroupsRequest.builder()
                   .maxResults(15)
                   .build();

           ListDatasetGroupsResponse groupsResponse =
personalizeClient.listDatasetGroups(groupsRequest);
           List<DatasetGroupSummary> groups = groupsResponse.datasetGroups();
           for (DatasetGroupSummary group : groups) {
               System.out.println("The DataSet name is : " + group.name());
               System.out.println("The DataSet ARN is : " +
group.datasetGroupArn());
           }

       } catch (PersonalizeException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous ListDatasetGroupsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`ListRecipes`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`ListRecipes`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public static void listAllRecipes(PersonalizeClient personalizeClient) {

        try {
            ListRecipesRequest recipesRequest = ListRecipesRequest.builder()
                    .maxResults(15)
                    .build();

            ListRecipesResponse response =
personalizeClient.listRecipes(recipesRequest);
            List<RecipeSummary> recipes = response.recipes();
            for (RecipeSummary recipe : recipes) {
                System.out.println("The recipe ARN is: " + recipe.recipeArn());
                System.out.println("The recipe name is: " + recipe.name());
            }
```

```
        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, voir ListRecipesla section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`ListSolutions`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`ListSolutions`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void listAllSolutions(PersonalizeClient personalizeClient,
String datasetGroupArn) {

        try {
            ListSolutionsRequest solutionsRequest =
ListSolutionsRequest.builder()
                    .maxResults(10)
                    .datasetGroupArn(datasetGroupArn)
                    .build();

            ListSolutionsResponse response =
personalizeClient.listSolutions(solutionsRequest);
```

```
            List<SolutionSummary> solutions = response.solutions();
            for (SolutionSummary solution : solutions) {
                System.out.println("The solution ARN is: " +
solution.solutionArn());
                System.out.println("The solution name is: " + solution.name());
            }

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, voir ListSolutionsla section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`UpdateCampaign`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`UpdateCampaign`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public static String updateCampaign(PersonalizeClient personalizeClient,
            String campaignArn,
            String solutionVersionArn,
            Integer minProvisionedTPS) {

        try {
```

```
            // build the updateCampaignRequest
            UpdateCampaignRequest updateCampaignRequest =
UpdateCampaignRequest.builder()
                    .campaignArn(campaignArn)
                    .solutionVersionArn(solutionVersionArn)
                    .minProvisionedTPS(minProvisionedTPS)
                    .build();

            // update the campaign
            personalizeClient.updateCampaign(updateCampaignRequest);

            DescribeCampaignRequest campaignRequest =
DescribeCampaignRequest.builder()
                    .campaignArn(campaignArn)
                    .build();

            DescribeCampaignResponse campaignResponse =
personalizeClient.describeCampaign(campaignRequest);
            Campaign updatedCampaign = campaignResponse.campaign();

            System.out.println("The Campaign status is " +
updatedCampaign.status());
            return updatedCampaign.status();

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, voir UpdateCampaignla section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Exemples de code pour Amazon Personalize Events à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Personalize Events avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Amazon Personalize Events

- Exemples de base pour Amazon Personalize Events utilisant AWS SDKs
  - Actions pour Amazon Personalize Events à l'aide de AWS SDKs
    - Utilisation PutEvents avec un AWS SDK
    - Utilisation PutItems avec un AWS SDK
    - Utilisation PutUsers avec un AWS SDK

Exemples de base pour Amazon Personalize Events utilisant AWS SDKs

Les exemples de code suivants montrent comment utiliser les principes de base d'Amazon Personalize Events avec AWS SDKs.

Exemples

- Actions pour Amazon Personalize Events à l'aide de AWS SDKs
  - Utilisation PutEvents avec un AWS SDK
  - Utilisation PutItems avec un AWS SDK
  - Utilisation PutUsers avec un AWS SDK

Actions pour Amazon Personalize Events à l'aide de AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Amazon Personalize Events individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir une liste complète, consultez la Référence des API Amazon Personalize Events.

Exemples

- Utilisation PutEvents avec un AWS SDK
- Utilisation PutItems avec un AWS SDK
- Utilisation PutUsers avec un AWS SDK

Utilisation **`PutEvents`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `PutEvents`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        public static int putItems(PersonalizeEventsClient
personalizeEventsClient,
                        String datasetArn,
                        String item1Id,
                        String item1PropertyName,
                        String item1PropertyValue,
                        String item2Id,
                        String item2PropertyName,
                        String item2PropertyValue) {

                int responseCode = 0;
```

```
                ArrayList<Item> items = new ArrayList<>();

                try {
                        Item item1 = Item.builder()
                                        .itemId(item1Id)
                                        .properties(String.format("{\"%1$s\":
\"%2$s\"}",
                                                        item1PropertyName,
item1PropertyValue))
                                        .build();

                        items.add(item1);

                        Item item2 = Item.builder()
                                        .itemId(item2Id)
                                        .properties(String.format("{\"%1$s\":
\"%2$s\"}",
                                                        item2PropertyName,
item2PropertyValue))
                                        .build();

                        items.add(item2);

                        PutItemsRequest putItemsRequest =
PutItemsRequest.builder()
                                        .datasetArn(datasetArn)
                                        .items(items)
                                        .build();

                        responseCode =
personalizeEventsClient.putItems(putItemsRequest).sdkHttpResponse().statusCode();
                        System.out.println("Response code: " + responseCode);
                        return responseCode;

                } catch (PersonalizeEventsException e) {
                        System.out.println(e.awsErrorDetails().errorMessage());
                }
                return responseCode;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { PutEventsCommand } from "@aws-sdk/client-personalize-events";
import { personalizeEventsClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeEventsClient = new PersonalizeEventsClient({ region:
 "REGION"});

// Convert your UNIX timestamp to a Date.
const sentAtDate = new Date(1613443801 * 1000); // 1613443801 is a testing value.
 Replace it with your sentAt timestamp in UNIX format.

// Set put events parameters.
const putEventsParam = {
  eventList: [
    /* required */
    {
      eventType: "EVENT_TYPE" /* required */,
      sentAt: sentAtDate /* required, must be a Date with js */,
      eventId: "EVENT_ID" /* optional */,
      itemId: "ITEM_ID" /* optional */,
    },
  ],
  sessionId: "SESSION_ID" /* required */,
  trackingId: "TRACKING_ID" /* required */,
  userId: "USER_ID" /* required */,
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutEventsCommand(putEventsParam),
    );
    console.log("Success!", response);
```

```
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`PutItems`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`PutItems`.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { PutItemsCommand } from "@aws-sdk/client-personalize-events";
import { personalizeEventsClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeEventsClient = new PersonalizeEventsClient({ region:
 "REGION"});

// Set the put items parameters. For string properties and values, use the \
 character to escape quotes.
const putItemsParam = {
  datasetArn: "DATASET_ARN" /* required */,
  items: [
```

```
    /* required */
    {
      itemId: "ITEM_ID" /*  required */,
      properties:
        '{"PROPERTY1_NAME": "PROPERTY1_VALUE", "PROPERTY2_NAME":
 "PROPERTY2_VALUE", "PROPERTY3_NAME": "PROPERTY3_VALUE"}' /* optional */,
    },
  ],
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutItemsCommand(putItemsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous PutItemsà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`PutUsers`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `PutUsers`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
       public static int putUsers(PersonalizeEventsClient
personalizeEventsClient,
                       String datasetArn,
                       String user1Id,
                       String user1PropertyName,
                       String user1PropertyValue,
                       String user2Id,
                       String user2PropertyName,
                       String user2PropertyValue) {

               int responseCode = 0;
               ArrayList<User> users = new ArrayList<>();

               try {
                       User user1 = User.builder()
                                       .userId(user1Id)
                                       .properties(String.format("{\"%1$s\":
\"%2$s\"}",
                                                       user1PropertyName,
user1PropertyValue))
                                       .build();

                       users.add(user1);

                       User user2 = User.builder()
                                       .userId(user2Id)
                                       .properties(String.format("{\"%1$s\":
\"%2$s\"}",
                                                       user2PropertyName,
user2PropertyValue))
                                       .build();
```

```
                        users.add(user2);

                        PutUsersRequest putUsersRequest =
PutUsersRequest.builder()
                                        .datasetArn(datasetArn)
                                        .users(users)
                                        .build();

                        responseCode =
personalizeEventsClient.putUsers(putUsersRequest).sdkHttpResponse().statusCode();
                        System.out.println("Response code: " + responseCode);
                        return responseCode;

                } catch (PersonalizeEventsException e) {
                        System.out.println(e.awsErrorDetails().errorMessage());
                }
                return responseCode;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutUsersà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { PutUsersCommand } from "@aws-sdk/client-personalize-events";
import { personalizeEventsClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeEventsClient = new PersonalizeEventsClient({ region:
 "REGION"});
```

```
// Set the put users parameters. For string properties and values, use the \
 character to escape quotes.
const putUsersParam = {
  datasetArn: "DATASET_ARN",
  users: [
    {
      userId: "USER_ID",
      properties: '{"PROPERTY1_NAME": "PROPERTY1_VALUE"}',
    },
  ],
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutUsersCommand(putUsersParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous PutUsersà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Exemples de code pour Amazon Personalize Runtime à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Personalize Runtime avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Amazon Personalize Runtime

- Exemples de base pour Amazon Personalize Runtime utilisant AWS SDKs
  - Actions pour Amazon Personalize Runtime à l'aide de AWS SDKs
    - Utilisation GetPersonalizedRanking avec un AWS SDK
    - Utilisation GetRecommendations avec un AWS SDK

Exemples de base pour Amazon Personalize Runtime utilisant AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Amazon Personalize Runtime avec AWS SDKs.

Exemples

- Actions pour Amazon Personalize Runtime à l'aide de AWS SDKs
  - Utilisation GetPersonalizedRanking avec un AWS SDK
  - Utilisation GetRecommendations avec un AWS SDK

Actions pour Amazon Personalize Runtime à l'aide de AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Amazon Personalize Runtime individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez la Référence des API Amazon Personalize Runtime.

Exemples

- Utilisation GetPersonalizedRanking avec un AWS SDK
- Utilisation GetRecommendations avec un AWS SDK

Utilisation **`GetPersonalizedRanking`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `GetPersonalizedRanking`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static List<PredictedItem> getRankedRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
           String campaignArn,
           String userId,
           ArrayList<String> items) {

       try {
           GetPersonalizedRankingRequest rankingRecommendationsRequest =
GetPersonalizedRankingRequest.builder()
                   .campaignArn(campaignArn)
                   .userId(userId)
                   .inputList(items)
                   .build();

           GetPersonalizedRankingResponse recommendationsResponse =
personalizeRuntimeClient
                   .getPersonalizedRanking(rankingRecommendationsRequest);
           List<PredictedItem> rankedItems =
recommendationsResponse.personalizedRanking();
           int rank = 1;
           for (PredictedItem item : rankedItems) {
               System.out.println("Item ranked at position " + rank + "
details");
               System.out.println("Item Id is : " + item.itemId());
               System.out.println("Item score is : " + item.score());

System.out.println("-----------------------------------------");
               rank++;
           }
```

```
            return rankedItems;
        } catch (PersonalizeRuntimeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetPersonalizedRankingà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { GetPersonalizedRankingCommand } from "@aws-sdk/client-personalize-
runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set the ranking request parameters.
export const getPersonalizedRankingParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  inputList: ["ITEM_ID_1", "ITEM_ID_2", "ITEM_ID_3", "ITEM_ID_4"],
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetPersonalizedRankingCommand(getPersonalizedRankingParam),
```

```
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous GetPersonalizedRankingà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetRecommendations`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `GetRecommendations`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez une liste d'éléments recommandés.

```
   public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
String campaignArn, String userId) {

       try {
           GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
                   .campaignArn(campaignArn)
```

```
                    .numResults(20)
                    .userId(userId)
                    .build();

            GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
                    .getRecommendations(recommendationsRequest);
            List<PredictedItem> items = recommendationsResponse.itemList();
            for (PredictedItem item : items) {
                System.out.println("Item Id is : " + item.itemId());
                System.out.println("Item score is : " + item.score());
            }

        } catch (AwsServiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

Obtenez une liste d'éléments recommandés à partir d'un outil de recommandation créé dans un groupe de jeux de données de domaine.

```
    public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
String recommenderArn,
            String userId) {

        try {
            GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
                    .recommenderArn(recommenderArn)
                    .numResults(20)
                    .userId(userId)
                    .build();

            GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
                    .getRecommendations(recommendationsRequest);
            List<PredictedItem> items = recommendationsResponse.itemList();

            for (PredictedItem item : items) {
                System.out.println("Item Id is : " + item.itemId());
                System.out.println("Item score is : " + item.score());
```

```
            }
        } catch (AwsServiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

Utilisez un filtre lorsque vous demandez des recommandations.

```
    public static void getFilteredRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
            String campaignArn,
            String userId,
            String filterArn,
            String parameter1Name,
            String parameter1Value1,
            String parameter1Value2,
            String parameter2Name,
            String parameter2Value) {

        try {

            Map<String, String> filterValues = new HashMap<>();

            filterValues.put(parameter1Name, String.format("\"%1$s\",\"%2$s\"",
                    parameter1Value1, parameter1Value2));
            filterValues.put(parameter2Name, String.format("\"%1$s\"",
                    parameter2Value));

            GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
                    .campaignArn(campaignArn)
                    .numResults(20)
                    .userId(userId)
                    .filterArn(filterArn)
                    .filterValues(filterValues)
                    .build();

            GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
                    .getRecommendations(recommendationsRequest);
            List<PredictedItem> items = recommendationsResponse.itemList();
```

```
            for (PredictedItem item : items) {
                System.out.println("Item Id is : " + item.itemId());
                System.out.println("Item score is : " + item.score());
            }
        } catch (PersonalizeRuntimeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetRecommendationsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";

import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
};

export const run = async () => {
```

```
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Obtenez des recommandations à l'aide d'un filtre (groupe de jeu de données personnalisé).

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  recommenderArn: "RECOMMENDER_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Obtenez des recommandations filtrées à partir d'un système de recommandation créé dans un groupe de jeux de données de domaine.

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here:
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
 "REGION"});

// Set recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN" /* required */,
  userId: "USER_ID" /* required */,
  numResults: 15 /* optional */,
  filterArn: "FILTER_ARN" /* required to filter recommendations */,
  filterValues: {
    PROPERTY:
      '"VALUE"' /* Only required if your filter has a placeholder parameter */,
  },
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(
      new GetRecommendationsCommand(getRecommendationsParam),
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous GetRecommendationsà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Personalize avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

La sécurité dans Amazon Personalize

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Amazon Personalize utilise le chiffrement des données pour protéger vos données. Pour plus d'informations, consultezChiffrement des données dans Amazon Personalize. Third-partyles auditeurs testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon Personalize, consultez la section AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon Personalize. Les rubriques suivantes expliquent comment configurer Amazon Personalize pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Amazon Personalize.

Rubriques

- Protection des données dans Amazon Personalize
- Identity and Access Management pour Amazon Personalize
- Surveillance d'Amazon Personalize avec Amazon CloudWatch
- Journalisation des appels d'API Amazon Personalize avec AWS CloudTrail
- Validation de conformité pour Amazon Personalize
- Résilience dans Amazon Personalize
- Sécurité de l'infrastructure dans Amazon Personalize

- Amazon Personalize et interface avec les points de terminaison VPC (AWS PrivateLink)

Protection des données dans Amazon Personalize

Le modèle de responsabilité AWS partagée (modèle de) s'applique à la protection des données dans Amazon Personalize. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la FAQ sur la confidentialité des données et les . Pour plus d'informations sur la protection des données en Europe, consultez le Centre du règlement général sur la protection des données (RGPD).

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que

le champ Nom. Cela inclut lorsque vous travaillez avec Amazon Personalize ou une autre entreprise Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement des données dans Amazon Personalize

Les informations suivantes expliquent les domaines dans lesquels Amazon Personalize utilise le chiffrement des données pour protéger vos données.

Chiffrement au repos

Toutes les données stockées dans Amazon Personalize sont toujours chiffrées au repos à l'aide des clés gérées par Amazon Personalize AWS Key Management Service (AWS KMS). Si vous fournissez votre propre AWS KMS clé lors de la création d'une ressource, Amazon Personalize l'utilise pour chiffrer vos données et les stocker. Par exemple, si vous fournissez un AWS KMS ARN dans le cadre de l'CreateDatasetGroupopération, Amazon Personalize utilise la clé pour chiffrer et stocker les données que vous importez dans tous les ensembles de données que vous créez dans ce groupe de jeux de données.

Vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Pour plus d'informations sur le chiffrement des données dans Amazon S3, consultez la section Protection des données à l'aide du chiffrement dans le guide de l'utilisateur d'Amazon Simple Storage Service. Pour plus d'informations sur la gestion de votre propre AWS KMS clé, consultez la section Gestion des clés dans le Guide du AWS Key Management Service développeur.

Chiffrement en transit

Amazon Personalize utilise le protocole TLS avec AWS des certificats pour chiffrer toutes les données envoyées à d'autres AWS services. Toute communication avec d'autres AWS services s'effectue via HTTPS, et les points de terminaison Amazon Personalize ne prennent en charge que les connexions sécurisées via HTTPS.

Amazon Personalize copie les données de votre compte et les traite dans un AWS système interne. Lors du traitement des données, Amazon Personalize chiffre les données à l'aide d'une AWS KMS clé Amazon Personalize ou de toute autre AWS KMS clé que vous fournissez.

Gestion des clés

AWS gère toutes les AWS KMS clés par défaut. Il est de votre responsabilité de gérer les AWS KMS clés que vous possédez. Vous devez autoriser Amazon Personalize et votre rôle de service Amazon Personalize IAM à utiliser votre clé. Pour plus d'informations, consultez Autoriser Amazon Personalize à utiliser votre AWS KMS clé.

Pour plus d'informations sur la gestion de votre propre AWS KMS clé, consultez la section Gestion des clés dans le Guide du AWS Key Management Service développeur.

Identity and Access Management pour Amazon Personalize

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources Amazon Personalize. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion de l'accès à l'aide de politiques
- Comment Amazon Personalize fonctionne avec IAM
- Cross-service prévention confuse des adjoints
- Identity-based exemples de politiques pour Amazon Personalize
- Résolution des problèmes liés à l'identité et à l'accès Amazon Personalize

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes liés à l'identité et à l'accès Amazon Personalize)
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir Comment Amazon Personalize fonctionne avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir Identity-based exemples de politiques pour Amazon Personalize)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations

d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Identity-based politiques

Identity-based les politiques sont des documents de politique d'autorisation JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Identity-based les politiques peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Resource-based politiques

Resource-based les politiques sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Resource-based les politiques sont des politiques intégrées qui se trouvent dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCP) : spécifient les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez Politiques de contrôle de service dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCP) : définissent les autorisations maximales disponibles pour les ressources de votre organisation. Pour plus d'informations, consultez Politiques de contrôle des ressources (RCP) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment Amazon Personalize fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon Personalize, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon Personalize.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon Personalize

Fonctionnalité IAM	Assistance Amazon Personalize
Identity-based politiques	Oui
Resource-based politiques	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACL	Non

Fonctionnalité IAM	Assistance Amazon Personalize
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles de service	Oui
Service-linked rôles	Non

Pour obtenir une vue d'ensemble de la façon dont Amazon Personalize et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Identity-based politiques relatives à Amazon Personalize

Prend en charge les politiques basées sur l'identité : oui

Identity-based les politiques sont des documents de politique d'autorisation JSON que vous pouvez associer à une identité, telle qu'un utilisateur IAM, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Identity-based exemples de politiques pour Amazon Personalize

Pour consulter des exemples de politiques basées sur l'identité Amazon Personalize, consultez. Identity-based exemples de politiques pour Amazon Personalize

Resource-based politiques au sein d'Amazon Personalize

Prend en charge les politiques basées sur les ressources : non

Resource-based les politiques sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Actions politiques pour Amazon Personalize

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions Amazon Personalize, consultez la section Actions définies par Amazon Personalize dans le Service Authorization Reference.

Les actions politiques dans Amazon Personalize utilisent le préfixe suivant avant l'action :

```
personalize
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "personalize:action1",
      "personalize:action2"
         ]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "personalize:Describe*"
```

Pour consulter des exemples de politiques basées sur l'identité Amazon Personalize, consultez. Identity-based exemples de politiques pour Amazon Personalize

Ressources relatives aux politiques pour Amazon Personalize

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de ressources Amazon Personalize et leurs ARN, consultez la section Ressources définies par Amazon Personalize dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon Personalize.

Pour consulter des exemples de politiques basées sur l'identité Amazon Personalize, consultez. Identity-based exemples de politiques pour Amazon Personalize

Clés relatives aux conditions de politique pour Amazon Personalize

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition Amazon Personalize, consultez la section Clés de condition pour Amazon Personalize dans le Service Authorization Reference. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon Personalize.

Pour consulter des exemples de politiques basées sur l'identité Amazon Personalize, consultez. Identity-based exemples de politiques pour Amazon Personalize

ACL dans Amazon Personalize

Prend en charge les ACL : non

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec Amazon Personalize

Prise en charge d'ABAC (balises dans les politiques) : Oui

Attribute-based le contrôle d'accès (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction d'attributs appelés balises. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur le balisage des ressources Amazon Personalize, consultezMarquer les ressources Amazon Personalize.

Pour visualiser un exemple de politique basée sur l'identité permettant de limiter l'accès à une ressource en fonction des balises de cette ressource, consultez Utilisation de balises dans les politiques IAM.

Utilisation d'informations d'identification temporaires avec Amazon Personalize

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM et Services AWS compatibles avec IAM dans le Guide de l'utilisateur IAM.

Cross-service autorisations principales pour Amazon Personalize

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section Sessions de transmission d'accès.

Rôles de service pour Amazon Personalize

Prend en charge les rôles de service : oui

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon Personalize. Modifiez les rôles de service uniquement lorsque Amazon Personalize fournit des conseils à cet effet.

Service-linked rôles pour Amazon Personalize

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut assumer le rôle d'effectuer une action en votre nom. Service-linked les rôles apparaissent dans votre fichier Compte AWS et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne des Service-linked rôles. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Cross-service prévention confuse des adjoints

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. Cross-service l'usurpation d'identité peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé à accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés contextuelles de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qu'Amazon Personalize accorde à un autre service à la ressource.

Pour éviter le problème de confusion lié aux rôles assumés par Amazon Personalize, définissez la valeur de `aws:SourceArn` à dans la politique de confiance du

rôle`arn:aws:personalize:region:accountNumber:*`. Le caractère générique (*) applique la condition à toutes les ressources Amazon Personalize.

La politique de relation de confiance suivante accorde à Amazon Personalize l'accès à vos ressources et utilise les clés de contexte `aws:SourceArn` et de condition `aws:SourceAccount` globale pour éviter le problème de confusion des adjoints. Utilisez cette politique lorsque vous créez un rôle pour Amazon Personalize (Création d'un rôle IAM pour Amazon Personalize).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "personalize.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
        "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
        "aws:SourceArn": "arn:aws:personalize:us-east-1:444455556666:*"
        }
      }
    }
  ]
}
```

Identity-based exemples de politiques pour Amazon Personalize

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources Amazon Personalize. Pour octroyer aux utilisateurs des autorisations d’effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon Personalize, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon Personalize dans le Service Authorization Reference.

Rubriques

- Bonnes pratiques en matière de politiques
- AWS stratégies gérées
- Utilisation de la console Amazon Personalize
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Permettre un accès complet aux ressources Amazon Personalize
- Autoriser l'accès en lecture seule aux ressources Amazon Personalize

Bonnes pratiques en matière de politiques

Identity-based les politiques déterminent si quelqu'un peut créer, accéder ou supprimer les ressources Amazon Personalize de votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.

- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

AWS stratégies gérées

AWS les politiques gérées sont des politiques créées et gérées par AWS. Vous trouverez ci-dessous des exemples de politiques AWS gérées que vous pouvez utiliser lorsque vous travaillez avec Amazon Personalize.

AmazonPersonalizeFullAccess Stratégie

Vous pouvez utiliser la `AmazonPersonalizeFullAccess` politique AWS gérée pour accorder aux utilisateurs les autorisations suivantes :

- Accédez à toutes les ressources Amazon Personalize
- Publier et répertorier les statistiques sur Amazon CloudWatch

- Répertorier, lire, écrire et supprimer tous les objets d'un compartiment Amazon S3 contenant `Personalize` ou figurant `personalize` dans le nom du compartiment
- Transmettre un rôle à Amazon Personalize

`AmazonPersonalizeFullAccess`fournit plus d'autorisations que nécessaire. Nous vous recommandons de créer une nouvelle politique IAM qui n'accorde que les autorisations nécessaires (voirAutoriser Amazon Personalize à accéder à vos ressources).

CloudWatchFullAccess

Pour autoriser vos utilisateurs à surveiller Amazon Personalize avec CloudWatch, associez la `CloudWatchFullAccess` politique à votre rôle. Pour plus d'informations, consultez Surveillance d'Amazon Personalize avec Amazon CloudWatch.

La `CloudWatchFullAccess` politique est facultative et autorise les actions suivantes :

- Publiez et listez les métriques Amazon Personalize dans CloudWatch
- Consultez les métriques et les statistiques des métriques.
- Définissez des alarmes basées sur des métriques.

Utilisation de la console Amazon Personalize

Pour accéder à la console Amazon Personalize, vous devez disposer d'un minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon Personalize présentes dans votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Permettre un accès complet aux ressources Amazon Personalize

L'exemple suivant donne à un utilisateur IAM de votre AWS compte un accès complet à toutes les ressources et actions Amazon Personalize.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Autoriser l'accès en lecture seule aux ressources Amazon Personalize

Dans cet exemple, vous accordez à un utilisateur IAM de votre AWS compte un accès en lecture seule à vos ressources Amazon Personalize, notamment aux ensembles de données Amazon Personalize, aux groupes de jeux de données, aux solutions et aux campagnes.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeAlgorithm",
                "personalize:DescribeBatchInferenceJob",
                "personalize:DescribeBatchSegmentJob",
                "personalize:DescribeCampaign",
                "personalize:DescribeDataset",
                "personalize:DescribeDatasetExportJob",
                "personalize:DescribeDatasetGroup",
                "personalize:DescribeDatasetImportJob",
                "personalize:DescribeEventTracker",
                "personalize:DescribeFeatureTransformation",
                "personalize:DescribeFilter",
                "personalize:DescribeRecipe",
                "personalize:DescribeRecommender",
                "personalize:DescribeSchema",
                "personalize:DescribeSolution",
                "personalize:DescribeSolutionVersion",
```

```
                "personalize:GetSolutionMetrics",
                "personalize:ListBatchInferenceJobs",
                "personalize:ListBatchSegmentJobs",
                "personalize:ListCampaigns",
                "personalize:ListDatasetExportJobs",
                "personalize:ListDatasetGroups",
                "personalize:ListDatasetImportJobs",
                "personalize:ListDatasets",
                "personalize:ListEventTrackers",
                "personalize:ListFilters",
                "personalize:ListRecipes",
                "personalize:ListRecommenders",
                "personalize:ListSchemas",
                "personalize:ListSolutions",
                "personalize:ListSolutionVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

Résolution des problèmes liés à l'identité et à l'accès Amazon Personalize

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon Personalize et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Amazon Personalize
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon Compte AWS pour accéder à mes ressources Amazon Personalize

Je ne suis pas autorisé à effectuer une action dans Amazon Personalize

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `personalize:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 personalize:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `personalize:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon Personalize.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon Personalize. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon Compte AWS pour accéder à mes ressources Amazon Personalize

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si Amazon Personalize prend en charge ces fonctionnalités, consultezComment Amazon Personalize fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Surveillance d'Amazon Personalize avec Amazon CloudWatch

Avec Amazon CloudWatch, vous pouvez obtenir des statistiques associées à Amazon Personalize. Vous pouvez configurer des alarmes pour vous avertir lorsqu'une ou plusieurs métriques dépassent un seuil défini. Pour consulter les statistiques, vous pouvez utiliser Amazon CloudWatch AWS Command Line Interface, Amazon ou l'CloudWatch API.

Rubriques

- Utilisation CloudWatch des métriques pour Amazon Personalize
- Accès aux métriques d'Amazon Personalize

- Création d'une alarme
- Exemple d'application de surveillance sans serveur Amazon Personalize
- CloudWatch statistiques pour Amazon Personalize

Utilisation CloudWatch des métriques pour Amazon Personalize

Pour utiliser les métriques, vous devez spécifier les informations suivantes :

- Nom de la métrique .
- La dimension de métrique. Une dimension est une paire nom-valeur qui vous aide à identifier une métrique de façon unique.

Vous pouvez obtenir des données de surveillance pour Amazon Personalize à l' Console de gestion AWS aide de l'API AWS CLI, de ou de l' CloudWatch API. Vous pouvez également utiliser l' CloudWatch API par le biais de l'un des AWS SDK ou des outils CloudWatch d'API. La console affiche une série de graphiques basés sur les données brutes de l' CloudWatch API. En fonction de vos besoins, vous pouvez utiliser les graphiques affichés dans la console ou extraits de l'API.

La liste suivante présente certaines utilisations courantes des métriques. Voici quelques suggestions pour vous aider à démarrer, qui ne forment pas une liste exhaustive.

Comment... ?	Métrique pertinente
Comment puis-je suivre le nombre d'événements qui ont été enregistrés ?	Surveiller la métrique `PutEventsRequests` .
Comment puis-je surveiller les DatasetImportJob erreurs ?	Utilisez la métrique `DatasetImportJobError` .
Comment puis-je surveiller la latence des appels `GetRecommendations` ?	Utilisez la métrique `GetRecommendationsLatency` .

Vous devez disposer des CloudWatch autorisations appropriées pour surveiller Amazon Personalize avec CloudWatch. Pour plus d'informations, consultez Authentification et contrôle d'accès pour Amazon CloudWatch.

Accès aux métriques d'Amazon Personalize

Les exemples suivants montrent comment accéder aux métriques Amazon Personalize à l'aide de la CloudWatch console AWS CLI, de l'API et de l' CloudWatch API.

Pour consulter les métriques (console)

1. Connectez-vous à la CloudWatch console Console de gestion AWS et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Métriques, cliquez sur l'onglet Toutes les mesures, puis sélectionnez`AWS/Personalize`.
3. Choisissez la dimension de métrique.
4. Choisissez la métrique souhaitée dans la liste, puis une période pour le graphique.

Pour afficher les métriques pour les événements reçus sur une période de temps (CLI)

- Ouvrez le AWS CLI et entrez la commande suivante :

```
aws cloudwatch get-metric-statistics \
  --metric-name PutEventsRequests \
  --start-time 2019-03-15T00:00:20Z \
  --period 3600 \
  --end-time 2019-03-16T00:00:00Z \
  --namespace AWS/Personalize \
  --dimensions Name=EventTrackerArn,Value=EventTrackerArn \
  --statistics Sum
```

  Cet exemple illustre les événements reçus pour l'ARN de dispositif de suivi d'événements donné sur une période de temps. Pour de plus amples informations, veuillez consulter get-metric-statistics.

Pour accéder aux métriques (CloudWatch API)

- Appelez `GetMetricStatistics`. Pour plus d'informations, consultez le Amazon CloudWatch API Reference.

Création d'une alarme

Vous pouvez créer une CloudWatch alarme qui envoie un message Amazon Simple Notification Service (Amazon SNS) lorsque l'état de l'alarme change. Une alarme surveille une métrique sur la période que vous spécifiez. Elle réalise une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. L'action est une notification envoyée à une rubrique ou à une AWS Auto Scaling politique Amazon SNS.

Les alarmes déclenchent des actions uniquement pour les changements d'état prolongés. CloudWatch les alarmes n'appellent pas d'actions simplement parce qu'elles sont dans un état particulier. L'état doit avoir changé et avoir été maintenu pendant un nombre de périodes spécifié.

Pour définir une alarme (console)

1. Connectez-vous à la CloudWatch console Console de gestion AWS et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Alarmes, puis Créer une alarme. L'assistant Create Alarm démarre.
3. Choisissez Select metric (Sélectionner une métrique).
4. Dans l'onglet Toutes les mesures, choisissez`AWS/Personalize`.
5. Choisissez EventTrackerArn, puis sélectionnez PutEventsRequestsles métriques.
6. Sélectionnez l'onglet Graphed metrics (Graphiques des métriques).
7. Pour Statistic (Statistique), choisissez Sum (Somme).
8. Choisissez Select metric (Sélectionner une métrique).
9. Remplissez les champs Nom et Description. Pour Lorsque, choisissez **>**, puis entrez une valeur maximale de votre choix.
10. Si vous souhaitez vous CloudWatch envoyer un e-mail lorsque l'état d'alarme est atteint, pour Whenever this alarm :, choisissez State is ALARM. Afin d'envoyer des alarmes à une rubrique Amazon SNS existante, pour Envoyer une notification à : choisissez une rubrique SNS existante. Pour définir le nom et les adresses e-mail d'une nouvelle liste d'abonnement par e-mail, choisissez Nouvelle liste. CloudWatch enregistre la liste et l'affiche sur le terrain afin que vous puissiez l'utiliser pour définir de futures alarmes.

> **Note**
>
> Si vous utilisez Nouvelle liste pour créer une nouvelle rubrique Amazon SNS, les adresses e-mail doivent être vérifiées avant que les destinataires ne reçoivent de notifications. Amazon SNS n'envoie les e-mails que lorsque l'alarme passe à l'état d'alarme. Si ce changement d'état de l'alarme se produit avant la vérification des adresses de messagerie, les destinataires prévus ne reçoivent pas de notification.

11. Sélectionnez Créer une alerte.

Pour régler une alarme (AWS CLI)

- Ouvrez le AWS CLI, puis entrez la commande suivante. Modifiez la valeur du `alarm-actions` paramètre pour faire référence à une rubrique Amazon SNS que vous avez créée précédemment.

```
aws cloudwatch put-metric-alarm \
    --alarm-name PersonalizeCLI \
    --alarm-description "Alarm when more than 10 events occur" \
    --metric-name PutEventsRequests \
    --namespace  AWS/Personalize \
    --statistic Sum \
    --period 300 \
    --threshold 10 \
    --comparison-operator GreaterThanThreshold \
    --evaluation-periods 1 \
    --unit Count \
    --dimensions Name=EventTrackerArn,Value=EventTrackerArn \
    --alarm-actions SNSTopicArn
```

  Cet exemple montre comment créer une alarme lorsque plus de 10 événements se produisent pour l'ARN de dispositif de suivi d'événement donné dans un délai de 5 minutes. Pour de plus amples informations, veuillez consulter put-metric-alarm.

Pour configurer une alarme (CloudWatch API)

- Appelez `PutMetricAlarm`. Pour plus d'informations, consultez Amazon CloudWatch API Reference.

Exemple d'application de surveillance sans serveur Amazon Personalize

Pour un exemple d'application qui ajoute des fonctionnalités de surveillance, d'alerte et d'optimisation pour Amazon Personalize, consultez Amazon Personalize monitor dans le référentiel d'échantillons Amazon Personalize.

CloudWatch statistiques pour Amazon Personalize

Cette section contient des informations sur les CloudWatch métriques Amazon disponibles pour Amazon Personalize. Pour plus d'informations, consultez Surveillance d'Amazon Personalize avec Amazon CloudWatch.

Le tableau suivant répertorie les métriques Amazon Personalize. Toutes les mesures sauf GetRecommendations et GetPersonalizedRanking soutiennent ces statistiques :`Average`, `Minimum`, `Maximum`, `Sum`. GetRecommendations et GetPersonalizedRanking support `Sum` uniquement.

Métrique	Description
`DatasetImportJobRequests`	Le nombre d'appels d'API CreateDatasetImportJob réussis. Dimensions : `DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn`
`DatasetImportJobError`	Le nombre d'appels d'API `CreateDatasetImportJob` ayant entraîné une erreur. Dimensions : `DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn`
`DatasetImportJobExecutionTime`	Le temps entre l'appel d'API `CreateDatasetImportJob` et l'achèvement (ou l'échec) de l'opération. Dimensions : `DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn` Unité : secondes

Métrique	Description
DatasetSize	La taille des données importées par la tâche d'importation d'ensembles de données. Dimensions : DatasetGroupArn, DatasetArn, DatasetImportJobArn Unité : octets
SolutionTrainingJobRequests	Le nombre d'appels d'API CreateSolutionVersion réussis. Dimensions : SolutionArn, SolutionVersionArn
SolutionTrainingJobError	Le nombre d'appels d'API CreateSolutionVersion ayant entraîné une erreur. Dimensions : SolutionArn, SolutionVersionArn
SolutionTrainingJobExecutionTime	Le temps entre l'appel d'API CreateSolutionVersion et l'achèvement (ou l'échec) de l'opération. Dimensions : SolutionArn, SolutionVersionArn Unité : secondes
GetPersonalizedRanking	Si un appel GetPersonalizedRanking d'API est réussi. Utilisez les sum statistiques pour afficher le nombre total d'appels d' GetPersonalizedRanking API réussis. Cette métrique ne prend pas en charge les autres statistiques. Dimension : CampaignArn
GetPersonalizedRanking4xxErrors	Le nombre d'appels d'API GetPersonalizedRanking ayant retourné un code de réponse HTTP 4xx. Dimension : CampaignArn

Métrique	Description
GetPersonalizedRanking5xxErrors	Le nombre d'appels d'API GetPersonalizedRanking ayant retourné un code de réponse HTTP 5xx. Dimension : CampaignArn
GetPersonalizedRankingLatency	La durée entre la réception de l'appel GetPersonalizedRanking d'API et l'envoi des recommandations (à l'exception des erreurs 4xx et 5xx). Dimension : CampaignArn Unité : millisecondes
GetRecommendations	Si un appel GetRecommendations d'API est réussi. Utilisez les sum statistiques pour afficher le nombre total d'appels d' GetRecommendations API réussis. Cette métrique ne prend pas en charge les autres statistiques. Dimension : CampaignArn
GetRecommendations4xxErrors	Le nombre d'appels d'API GetRecommendations ayant retourné un code de réponse HTTP 4xx. Dimension : CampaignArn
GetRecommendations5xxErrors	Le nombre d'appels d'API GetRecommendations ayant retourné un code de réponse HTTP 5xx. Dimension : CampaignArn
GetRecommendationsLatency	La durée entre la réception de l'appel GetRecommendations d'API et l'envoi des recommandations (à l'exception des erreurs 4xx et 5xx). Dimension : CampaignArn Unité : millisecondes

Métrique	Description
PutEventsRequests	Le nombre d'appels d'API PutEvents réussis. Dimension : DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents4xxErrors	Le nombre d'appels d'API PutEvents ayant retourné un code de réponse HTTP 4xx. Dimension : DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents5xxErrors	Le nombre d'appels d'API PutEvents ayant retourné un code de réponse HTTP 5xx. Dimension : DatasetGroupArn, DatasetArn, EventTrackerArn
PutEventLatency	La durée nécessaire pour l'achèvement de l'appel PutEvents d'API (à l'exception des erreurs 4xx et 5xx). Dimension : DatasetGroupArn, DatasetArn, EventTrackerArn Unité : millisecondes
PutItemsRequests	Le nombre d'appels d'API PutItems réussis. Dimension : DatasetGroupArn, DatasetArn
PutItems4xxErrors	Le nombre d'appels d'API PutItems ayant retourné un code de réponse HTTP 4xx. Dimension : DatasetGroupArn, DatasetArn
PutItems5xxErrors	Le nombre d'appels d'API PutItems ayant retourné un code de réponse HTTP 5xx. Dimension : DatasetGroupArn, DatasetArn

Métrique	Description
`PutItemsLatency`	La durée nécessaire pour l'achèvement de l'appel `PutItems` d'API (à l'exception des erreurs 4xx et 5xx). Dimension : `DatasetGroupArn, DatasetArn` Unité : millisecondes
`PutUsersRequests`	Le nombre d'appels d'API PutUsers réussis. Dimension : `DatasetGroupArn, DatasetArn`
`PutUsers4xxErrors`	Le nombre d'appels d'API `PutUsers` ayant retourné un code de réponse HTTP 4xx. Dimension : `DatasetGroupArn, DatasetArn`
`PutUsers5xxErrors`	Le nombre d'appels d'API `PutUsers` ayant retourné un code de réponse HTTP 5xx. Dimension : `DatasetGroupArn, DatasetArn`
`PutUsersLatency`	La durée nécessaire pour l'achèvement de l'appel `PutUsers` d'API (à l'exception des erreurs 4xx et 5xx). Dimension : `DatasetGroupArn, DatasetArn` Unité : millisecondes

Journalisation des appels d'API Amazon Personalize avec AWS CloudTrail

Amazon Personalize est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Amazon Personalize. CloudTrail capture un sous-ensemble d'appels d'API pour Amazon Personalize sous forme d'événements, y compris les appels depuis la console Amazon Personalize et les appels de code vers les API Amazon Personalize. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon

Personalize. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Personalize, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le guide de AWS CloudTrail l'utilisateur.

Amazon Personalize les informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Personalize, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour Amazon Personalize, créez un historique. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s’applique à toutes les régions . Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d’informations, consultez :

- Présentation de la création d’un journal d’activité
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Amazon Personalize prend en charge l'enregistrement de chaque action (opération d'API) sous forme d'événement dans des fichiers CloudTrail journaux. Pour plus d’informations, consultez Actions.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l’identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Exemple : entrées du fichier journal Amazon Personalize

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal contenant des actions pour l'opération ListDatasetGroups d'API. Notez que l'opération d' ListDatasetGroups API étant une action qui ne change pas d'état, la `responseElements` réponse est nulle. Pour plus d'informations sur le corps des CloudTrail enregistrements, consultez la section Contenu des CloudTrail enregistrements.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "principal-id",
        "arn": "arn:aws:iam::user-arn",
        "accountId": "account-id",
        "accessKeyId": "access-key",
        "userName": "user-name"
    },
    "eventTime": "2018-11-22T02:18:03Z",
    "eventSource": "personalize.amazonaws.com",
    "eventName": "ListDatasetGroups",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "source-ip-address",
    "userAgent": "aws-cli/1.11.16 Python/2.7.11 Darwin/15.6.0 botocore/1.4.73",
    "requestParameters": null,
    "responseElements": null,
```

```
    "requestID": "request-id",
    "eventID": "event-id",
    "eventType": "AwsApiCall",
    "recipientAccountId": "recipient-account-id"
}
```

Validation de conformité pour Amazon Personalize

Third-party les auditeurs évaluent la sécurité et la conformité d'Amazon Personalize dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, HIPAA.

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformitéAWS . Pour des informations générales, consultez les programmes de AWS conformité, les programmes AWS .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Lorsque vous utilisez Amazon Personalize, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité Guides sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Découvrez comment vous pouvez l'utiliser AWS pour exécuter des charges de travail sensibles réglementées par le. U.S Health Insurance Portability and Accountability Act (HIPAA)
- AWS ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles énoncées dans le guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub CSPM— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans Amazon Personalize

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Amazon Personalize tire parti de l'infrastructure AWS mondiale pour la résilience des données. Lorsque vous créez une ressource Amazon Personalize dans une AWS région, Amazon Personalize gère la résilience et la redondance des données de la ressource dans plusieurs zones de disponibilité. Pour obtenir la liste des AWS régions dans lesquelles vous pouvez créer des ressources Amazon Personalize, consultez AWS les régions et les points de terminaison dans le manuel Amazon Web Services General Reference. Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Sécurité de l'infrastructure dans Amazon Personalize

En tant que service géré, Amazon Personalize est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Personalize via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Des suites de chiffrement dotées d'un secret de transmission parfait (PFS), telles que DHE (Ephemeral) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Amazon Personalize et interface avec les points de terminaison VPC (AWS PrivateLink)

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos AWS ressources, vous pouvez établir une connexion privée entre votre VPC et Amazon Personalize. Cette connexion permet à Amazon Personalize de communiquer avec vos ressources sur votre VPC sans passer par l'Internet public.

Amazon VPC est un système Service AWS que vous utilisez pour lancer AWS des ressources dans un cloud privé virtuel (VPC) ou un réseau virtuel que vous définissez. Avec un VPC, vous contrôlez des paramètres réseau, tels que la plage d'adresses IP, les sous-réseaux, les tables de routage et les passerelles réseau. Avec les points de terminaison VPC, le AWS réseau gère le routage entre votre VPC et. Services AWS

Pour connecter votre VPC à Amazon Personalize, vous devez définir un point de terminaison VPC d'interface pour Amazon Personalize. Un point de terminaison d'interface est une interface elastic network dotée d'une adresse IP privée qui sert de point d'entrée pour le trafic destiné à un opérateur pris en charge Service AWS. Le point de terminaison fournit une connectivité fiable et évolutive à Amazon Personalize. Il ne nécessite pas de passerelle Internet, d'instance de traduction d'adresses réseau (NAT) ou de connexion VPN. Pour de plus amples informations, consultez Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

Les points de terminaison VPC de l'interface sont activés par. AWS PrivateLink Cette AWS technologie permet une communication privée entre Services AWS eux en utilisant une interface elastic network avec des adresses IP privées.

> **Note**
>
> Tous les points de terminaison Amazon Personalize Federal Information Processing Standard (FIPS) sont pris en charge par. AWS PrivateLink

Rubriques

- Création d'un point de terminaison VPC d'interface pour Amazon Personalize
- Création d'une politique de point de terminaison VPC pour Amazon Personalize

Création d'un point de terminaison VPC d'interface pour Amazon Personalize

Vous pouvez créer un point de terminaison VPC pour le service Amazon Personalize à l'aide de la console Amazon VPC ou du (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez la section Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface dans le guide de l'utilisateur Amazon VPC.

Pour créer un point de terminaison VPC pour Amazon Personalize, choisissez l'une des options suivantes pour le service. Vous pouvez choisir de créer un point de terminaison ipv4, ipv6 ou dualstack.

- com.amazonaws. *region*.personnaliser
- com.amazonaws. *region*.personnalisez les événements
- com.amazonaws. *region*.personalize-runtime

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d'API à Amazon Personalize en utilisant son nom DNS par défaut pour la région, par exemple,`personalize.us-east-1.api.aws`.

Création d'une politique de point de terminaison VPC pour Amazon Personalize

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à Amazon Personalize. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour plus d’informations, consultez Contrôle de l’accès aux services avec points de terminaison d’un VPC dans le Guide de l’utilisateur Amazon VPC.

Exemple : politique de point de terminaison VPC autorisant toutes les actions Amazon Personalize et les actions PassRole

Lorsqu'elle est attachée à un point de terminaison, cette politique donne accès à toutes les actions Amazon Personalize et PassRole.

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:*",
                "iam:PassRole"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemple : politique de point de terminaison VPC autorisant les actions Amazon Personalize ListDatasets

Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux ListDatasets actions Amazon Personalize répertoriées.

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:ListDatasets"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Personalize : points de terminaison et quotas

Les sections suivantes contiennent des informations sur les directives, les quotas et les points de terminaison d'Amazon Personalize. Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Pour de plus amples informations, veuillez consulter Demande d'augmentation de quota.

Rubriques

- Amazon Personalize : points de terminaison et régions
- Conformité d'
- Quotas de service
- Demande d'augmentation de quota

Amazon Personalize : points de terminaison et régions

Pour obtenir la liste des points de terminaison Amazon Personalize par région, consultez la section AWS Régions et points de terminaison dans le manuel Amazon Web Services General Reference.

Conformité d'

Pour plus d'informations sur les programmes de conformité Amazon Personalize, consultez AWSAWS la section Conformité, programmes de conformité et AWS services concernés par programme de conformité.

Quotas de service

Les quotas de votre AWS compte pour Amazon Personalize sont les suivants.

Ressource	Quota
Item interactions	
Nombre minimum d'interactions avec des objets uniques requises pour créer une version de solution ou un recommandateur. Pour une	1 000

Ressource	Quota
solution personnalisée, vous devez disposer de ce nombre d'enregistrements après tout filtrage par type d'événement ou par valeur d'événement avant l'entraînement.	
Pour User-Personalization-v 2 et Personalized-Ranking-v 2 recettes, le nombre maximum d'interactions entre objets prises en compte par un modèle pendant l'entraînement.	3 milliards
Pour tous les cas d'utilisation du domaine et les recettes personnalisées autres que User-Personalization-v Personalized-Ranking-v 2 ou 2, le nombre maximum d'interactions entre objets prises en compte par un modèle pendant l'entraînement.	500 millions (ajustable)
Nombre maximal de types d'événements distincts combiné au nombre total de colonnes de métadonnées facultatives dans un jeu de données d'interactions entre éléments.	10
Nombre maximal de colonnes de métadonnées, à l'exception des champs réservés, dans un jeu de données d'interactions entre éléments.	5
Nombre maximal de caractères pour les données catégorielles et les valeurs d'impression.	1 000
Quantité maximale de données d'interactions entre éléments en masse par tâche d'importation de jeu de données avec le mode d'importation COMPLÈTE.	100 Go (passe à 1 To avec toute augmentation du nombre d'interactions entre objets prise en compte par un modèle)

Ressource	Quota
Quantité maximale de données d'interactions entre éléments en masse par tâche d'importation de jeu de données avec le mode d'importation INCRÉMENTAL.	1 Go
Nombre minimal d'enregistrements d'interactions entre éléments par tâche d'importation d'un ensemble de données avec le mode d'importation COMPLET ou INCRÉMENTAL.	1 000
Users	
Nombre minimum d'utilisateurs uniques dans les données d'interactions entre articles, avec au moins 2 interactions entre éléments chacun, requis pour créer un recommandeur de domaine ou une version de solution personnalisée.	25
Pourcentage minimum du nombre total d'utilisateurs qui doivent avoir au moins 2 interactions avec des éléments avant de pouvoir créer une recommandation de domaine ou une version de solution personnalisée.	1 pour cent
Nombre maximal de champs de métadonnées pour un ensemble de données utilisateurs.	25
Nombre maximum de caractères pour les valeurs de données USER_ID.	256
Nombre maximal de caractères pour les valeurs de données catégorielles.	1 000 caractères

Ressource	Quota
Quantité maximale de données utilisateur en masse par tâche d'importation de jeu de données avec le mode d'importation COMPLÈTE.	100 Go
Quantité maximale de données utilisateur en masse par tâche d'importation de jeu de données avec le mode d'importation INCRÉMENTAL.	1 Go
Items	
Pour User-Personalization-v 2 ou Personalized-Ranking-v 2, le nombre maximum d'éléments pris en compte par un modèle pendant l'entraîn ement. Ces éléments proviennent à la fois du jeu de données Items et du jeu de données des interactions Item.	5 millions
Pour tous les cas d'utilisation du domaine et les recettes personnalisées autres que User-Personalization-v 2 et Personalized-Ranking-v 2, le nombre maximum d'éléments pris en compte par un modèle lors de la formation et de la génération de recommandations.	750 000
Nombre maximal de champs de métadonnées pour un jeu de données Items.	100
Nombre maximum de caractères pour les valeurs de données ITEM_ID.	256
Nombre maximal de caractères pour les valeurs de données de chaîne catégoriques et non catégoriques.	1 000 caractères

Ressource	Quota
Nombre maximal de champs textuels pour un jeu de données Items.	1
Nombre maximal de caractères pour les valeurs de données textuelles pour les langues chinoise et japonaise.	7 000 caractères
Nombre maximal de caractères pour les valeurs de données textuelles pour toutes les autres langues.	20 000 caractères
Quantité maximale de données d'éléments en vrac par tâche d'importation de jeu de données avec le mode d'importation en bloc.	100 Go
Quantité maximale de données d'éléments en vrac par tâche d'importation de jeu de données avec le mode d'importation INCRÉMENTAL.	1 Go
Actions	
Nombre maximal d'actions prises en compte par un modèle lors de la formation et de la génération de recommandations.	1 000
Nombre maximal de champs de métadonnées pour un jeu de données Actions.	10
Nombre maximal de caractères pour les valeurs de données ACTION_ID.	256
Nombre maximal de caractères pour les valeurs de données catégorielles.	1 000 caractères
Quantité maximale de données d'actions groupées par tâche d'importation de jeu de données avec le mode d'importation en bloc.	100 Go

Ressource	Quota
Quantité maximale de données d'actions groupées par tâche d'importation de jeu de données avec le mode d'importation INCRÉMENTAL.	1 Go
Action interactions	
Nombre maximal d'interactions entre actions prises en compte par un modèle pendant l'entraînement.	500 millions
Nombre maximal de colonnes de métadonnées, à l'exception des champs réservés, dans un jeu de données Action interactions.	5
Quantité maximale de données d'interac tions en masse par tâche d'importation de jeu de données avec le mode d'importation COMPLÈTE.	100 Go (passe à 1 To avec toute augmentation des interactions entre éléments d'action prise en compte par un modèle)
Quantité maximale de données d'interac tions en masse par tâche d'importation de jeu de données avec le mode d'importation INCRÉMENTAL.	1 Go
Individual record import APIs	
Taux maximal de `PutEvents` demandes par groupe de jeux de données.	1000/seconde
Nombre maximal d'événements dans un appel `PutEvents` .	10
Taille maximale d'un événement.	10 Ko

Ressource	Quota
Taux maximal de `PutActionInteractions` demandes par groupe de jeux de données.	1000/seconde
Nombre maximal d'événements d'interaction par action dans un `PutActionInteractions` appel.	10
Taille maximale d'un événement d'interaction action-action.	10 Ko
Taux maximal de `PutItems` demandes par groupe de jeux de données.	10/seconde
Nombre maximum d'éléments par `PutItems` appel.	10
Taux maximal de `PutUsers` demandes par groupe de jeux de données.	10/seconde
Nombre maximum d'utilisateurs par `PutUsers` appel.	10
Taux maximal de `PutActions` demandes par groupe de jeux de données.	10/seconde
Nombre maximum d'utilisateurs par `PutActions` appel.	10
Legacy recipes	
Quantité maximale de données combinées pour les ensembles de données Users et Items pour les métadonnées HRNN et les recettes HRNN-ColdStart.	5 Go

Ressource	Quota
Nombre maximum d'éléments de démarrage à froid pris en charge par la recette HRNN-ColdStart pour entraîner un modèle (créer une version de solution).	80000
Nombre minimum d'éléments de démarrage à froid nécessaires à la recette HRNN-ColdStart pour entraîner un modèle (créer une version de solution).	100
Filters	
Nombre total de filtres par groupe de jeux de données.	30 (réglable)
Nombre maximal de champs de jeu de données distincts pour un filtre.	10
Nombre total de champs de jeu de données distincts dans tous les filtres d'un groupe de jeux de données.	20
Nombre maximum d'interactions avec des éléments par utilisateur et par type d'événement pris en compte par un filtre.	100 interactions (réglables)
Nombre maximal d'interactions par utilisateur et par type d'événement considéré par un filtre.	300 interactions d'action (réglables)
GetRecommendations / GetPersonalizedRanking / GetActionRecommendations requests	
Taux de transaction maximal pour `GetRecommendations` `GetActionRecommendations` et `GetPersonalizedRanking` demandes.	2500/sec

Ressource	Quota
Nombre maximal de demandes `GetRecommendations` par seconde par campagne.	500/sec
Nombre maximal de demandes `GetActionRecommendations` par seconde par campagne.	500/sec
Nombre maximal de demandes `GetPersonalizedRanking` par seconde par campagne.	500 par seconde.
Nombre maximum de colonnes de métadonnées par `GetRecommendations` `GetPersonalizedRanking` demande.	10
Nombre maximal de résultats de recommandation pour une `GetRecommendation` demande sans métadonnées.	500
Nombre maximal de résultats de recommandation pour une `GetRecommendation` demande contenant des métadonnées.	50
Nombre maximum d'éléments à classer dans une `GetPersonalizedRanking` demande sans métadonnées.	500
Nombre maximum d'éléments à classer dans une `GetPersonalizedRanking` demande contenant des métadonnées.	50
Metric attribution quotas	
Nombre maximum de métriques pour une attribution de métrique	10

Ressource	Quota
Nombre maximum de sources d'attribution d'événements uniques	100
Batch inference jobs	
Nombre maximal de fichiers d'entrée pour une tâche d'inférence par lots.	1 000
Taille maximale d'une entrée de tâche d'inférence par lots.	1 Go
Nombre maximum d'enregistrements par fichier d'entrée pour une tâche d'inférence par lots sans thèmes.	50 millions
Nombre maximum d'enregistrements par fichier d'entrée pour une tâche d'inférence par lots avec des thèmes.	100
Batch segment jobs	
Nombre maximal de fichiers d'entrée pour une tâche de segmentation par lots.	1 000
Taille maximale de la saisie de la tâche par segment par lots.	1 Go
Nombre maximum de requêtes par fichier d'entrée pour la recette Item-Affinity.	500
Nombre maximum de requêtes par fichier d'entrée pour la Item-Attribute-Affinity recette.	10
Nombre maximum d'utilisateurs par segment	5 millions
Data deletion jobs	

Ressource	Quota
Nombre maximal de tâches de suppression de données pour un groupe de jeux de données dont le statut est EN ATTENTE.	5 (réglable)
Taille totale maximale de votre fichier ou de vos fichiers d'entrée de suppression de données	100 Mo

Les quotas de votre AWS compte sont les suivants pour chaque région.

Ressource	Quota
Nombre total de schémas actifs.	500
Nombre total de groupes de jeu de données actifs.	5 (réglable)
Nombre total de tâches d'importation de jeux de données en attente ou en cours.	5
Nombre total de tâches d'inférence par lots en attente ou en cours.	5 (réglable)
Nombre total de tâches segmentées par lots en attente ou en cours.	5
Nombre total de versions de solution en attente ou en cours de développement.	20 (réglable)

Chaque groupe de jeux de données possède les quotas suivants.

Ressource	Quota
Nombre total de solutions actives.	10 (réglable)
Nombre total de campagnes actives.	5 (réglable)
Nombre total de recommandateurs.	5
Nombre total de filtres.	30 (réglable)

Ressource	Quota
Nombre total de champs de jeu de données distincts pour tous les filtres.	20
Nombre total de tâches de suppression de données pour un groupe de jeux de données dont le statut est EN ATTENTE.	5

Demande d’augmentation de quota

Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Les quotas Amazon Personalize suivants sont ajustables :

- Nombre maximum d'interactions entre objets prises en compte par un modèle pendant l'entraînement.
- Campagnes actives par groupe de jeux de données
- Groupes de jeux de données actifs
- Filtres actifs par groupe de jeux de données
- Solutions actives par groupe de jeux de données
- Quantité de données par importation incrémentielle
- Nombre maximum d'interactions avec un élément par utilisateur et par type d'événement pris en compte par un filtre
- Nombre total de tâches d'inférence par lots en attente ou en cours
- Nombre total de versions de solution en attente ou en cours
- Taux maximum de `PutEvents` `PutActionInteraction` demandes

Pour demander une augmentation de quota, utilisez la console Service Quotas et suivez les étapes décrites dans la section Demande d'augmentation de quota du Guide de l'utilisateur des Quotas de Service.

Référence des API

Cette section fournit de la documentation sur les opérations de l'API Amazon Personalize. Pour obtenir la liste des points de terminaison Amazon Personalize par région, consultez la section AWS Régions et points de terminaison dans le manuel de référence AWS général.

Rubriques

- Actions
- Types de données
- Types d'erreurs courants
- Paramètres communs

Actions

Les actions suivantes sont prises en charge par Amazon Personalize :

- CreateBatchInferenceJob
- CreateBatchSegmentJob
- CreateCampaign
- CreateDataDeletionJob
- CreateDataset
- CreateDatasetExportJob
- CreateDatasetGroup
- CreateDatasetImportJob
- CreateEventTracker
- CreateFilter
- CreateMetricAttribution
- CreateRecommender
- CreateSchema
- CreateSolution
- CreateSolutionVersion
- DeleteCampaign

- DeleteDataset
- DeleteDatasetGroup
- DeleteEventTracker
- DeleteFilter
- DeleteMetricAttribution
- DeleteRecommender
- DeleteSchema
- DeleteSolution
- DescribeAlgorithm
- DescribeBatchInferenceJob
- DescribeBatchSegmentJob
- DescribeCampaign
- DescribeDataDeletionJob
- DescribeDataset
- DescribeDatasetExportJob
- DescribeDatasetGroup
- DescribeDatasetImportJob
- DescribeEventTracker
- DescribeFeatureTransformation
- DescribeFilter
- DescribeMetricAttribution
- DescribeRecipe
- DescribeRecommender
- DescribeSchema
- DescribeSolution
- DescribeSolutionVersion
- GetSolutionMetrics
- ListBatchInferenceJobs
- ListBatchSegmentJobs
- ListCampaigns

- ListDataDeletionJobs
- ListDatasetExportJobs
- ListDatasetGroups
- ListDatasetImportJobs
- ListDatasets
- ListEventTrackers
- ListFilters
- ListMetricAttributionMetrics
- ListMetricAttributions
- ListRecipes
- ListRecommenders
- ListSchemas
- ListSolutions
- ListSolutionVersions
- ListTagsForResource
- StartRecommender
- StopRecommender
- StopSolutionVersionCreation
- TagResource
- UntagResource
- UpdateCampaign
- UpdateDataset
- UpdateMetricAttribution
- UpdateRecommender
- UpdateSolution

Les actions suivantes sont prises en charge par Amazon Personalize Events :

- PutActionInteractions
- PutActions
- PutEvents

- PutItems
- PutUsers

Les actions suivantes sont prises en charge par Amazon Personalize Runtime :

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

Amazon Personalize

Les actions suivantes sont prises en charge par Amazon Personalize :

- CreateBatchInferenceJob
- CreateBatchSegmentJob
- CreateCampaign
- CreateDataDeletionJob
- CreateDataset
- CreateDatasetExportJob
- CreateDatasetGroup
- CreateDatasetImportJob
- CreateEventTracker
- CreateFilter
- CreateMetricAttribution
- CreateRecommender
- CreateSchema
- CreateSolution
- CreateSolutionVersion
- DeleteCampaign
- DeleteDataset
- DeleteDatasetGroup
- DeleteEventTracker

- DeleteFilter
- DeleteMetricAttribution
- DeleteRecommender
- DeleteSchema
- DeleteSolution
- DescribeAlgorithm
- DescribeBatchInferenceJob
- DescribeBatchSegmentJob
- DescribeCampaign
- DescribeDataDeletionJob
- DescribeDataset
- DescribeDatasetExportJob
- DescribeDatasetGroup
- DescribeDatasetImportJob
- DescribeEventTracker
- DescribeFeatureTransformation
- DescribeFilter
- DescribeMetricAttribution
- DescribeRecipe
- DescribeRecommender
- DescribeSchema
- DescribeSolution
- DescribeSolutionVersion
- GetSolutionMetrics
- ListBatchInferenceJobs
- ListBatchSegmentJobs
- ListCampaigns
- ListDataDeletionJobs
- ListDatasetExportJobs
- ListDatasetGroups

- ListDatasetImportJobs
- ListDatasets
- ListEventTrackers
- ListFilters
- ListMetricAttributionMetrics
- ListMetricAttributions
- ListRecipes
- ListRecommenders
- ListSchemas
- ListSolutions
- ListSolutionVersions
- ListTagsForResource
- StartRecommender
- StopRecommender
- StopSolutionVersionCreation
- TagResource
- UntagResource
- UpdateCampaign
- UpdateDataset
- UpdateMetricAttribution
- UpdateRecommender
- UpdateSolution

CreateBatchInferenceJob

Service : Amazon Personalize

Génère des recommandations par lots sur la base d'une liste d'articles ou d'utilisateurs stockés dans Amazon S3 et exporte les recommandations vers un compartiment Amazon S3.

Pour générer des recommandations par lots, spécifiez l'ARN d'une version de solution et un URI Amazon S3 pour les données d'entrée et de sortie. Pour la personnalisation des utilisateurs, les articles populaires et les solutions de classement personnalisées, la tâche d'inférence par lots génère une liste d'éléments recommandés pour chaque identifiant utilisateur dans le fichier d'entrée. Pour les solutions d'articles connexes, le job génère une liste d'éléments recommandés pour chaque ID d'article dans le fichier d'entrée.

Pour plus d'informations, consultez la section Création d'une tâche d'inférence par lots.

Si vous utilisez la Similar-Items recette, Amazon Personalize peut ajouter des thèmes descriptifs aux recommandations par lots. Pour générer des thèmes, définissez le mode de la tâche sur `THEME_GENERATION` et spécifiez le nom du champ qui contient les noms des éléments dans les données d'entrée.

Pour plus d'informations sur la génération de thèmes, voir Recommandations par lots avec des thèmes issus de Content Generator.

Vous ne pouvez pas obtenir de recommandations par lots avec les Next-Best-Action recettes Trending-Now ou.

Syntaxe de la requête

```
{
   "batchInferenceJobConfig": {
      "itemExplorationConfig": {
         "string" : "string"
      },
      "rankingInfluence": {
         "string" : number
      }
   },
   "batchInferenceJobMode": "string",
   "filterArn": "string",
   "jobInput": {
      "s3DataSource": {
```

```
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "jobName": "string",
   "jobOutput": {
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "numResults": number,
   "roleArn": "string",
   "solutionVersionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ],
   "themeGenerationConfig": {
      "fieldsForThemeGeneration": {
         "itemName": "string"
      }
   }
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

batchInferenceJobConfig

Les détails de configuration d'une tâche d'inférence par lots.

Type : objet BatchInferenceJobConfig

Obligatoire : non

batchInferenceJobMode

Mode de la tâche d'inférence par lots. Pour générer des thèmes descriptifs pour des groupes d'éléments similaires, définissez le mode de travail surTHEME_GENERATION. Si vous ne souhaitez pas générer de thèmes, utilisez le thème par défautBATCH_INFERENCE.

Lorsque vous recevez des recommandations par lots avec des thèmes, vous devrez payer des frais supplémentaires. Pour plus d'informations, consultez la page Amazon Personalize pricing.

Type : Chaîne

Valeurs valides : `BATCH_INFERENCE | THEME_GENERATION`

Obligatoire : non

filterArn

L'ARN du filtre à appliquer à la tâche d'inférence par lots. Pour plus d'informations sur l'utilisation des filtres, consultez la section Recommandations relatives au filtrage par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

jobInput

Le chemin Amazon S3 qui mène au fichier d'entrée sur lequel baser vos recommandations. Le matériel d'entrée doit être au format JSON.

Type : objet BatchInferenceJobInput

Obligatoire : oui

jobName

Nom de la tâche d'inférence par lots à créer.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

jobOutput

Le chemin d'accès au compartiment Amazon S3 dans lequel la sortie de la tâche sera stockée.

Type : objet BatchInferenceJobOutput

Obligatoire : oui

numResults

Le nombre de recommandations à récupérer.

Type : Integer

Obligatoire : non

roleArn

L'ARN du rôle Amazon Identity and Access Management qui est autorisé à lire et à écrire dans vos compartiments Amazon S3 d'entrée et de sortie respectivement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : oui

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution qui sera utilisée pour générer les recommandations d'inférence par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Liste des balises à appliquer à la tâche d'inférence par lots.

Type : tableau d’objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

themeGenerationConfig

Pour les tâches de génération de thèmes, spécifiez le nom de la colonne de votre jeu de données Items qui contient le nom de chaque élément.

Type : objet ThemeGenerationConfig

Obligatoire : non

Syntaxe de la réponse

```
{
   "batchInferenceJobArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchInferenceJobArn

L'ARN de la tâche d'inférence par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d’état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateBatchSegmentJob

Service : Amazon Personalize

Crée une tâche segmentée par lots. L'opération peut gérer jusqu'à 50 millions d'enregistrements et le fichier d'entrée doit être au format JSON. Pour plus d'informations, voir Obtenir des recommandations par lots et des segments d'utilisateurs.

Syntaxe de la requête

```
{
   "filterArn": "string",
   "jobInput": {
      "s3DataSource": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "jobName": "string",
   "jobOutput": {
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "numResults": number,
   "roleArn": "string",
   "solutionVersionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

filterArn

L'ARN du filtre à appliquer à la tâche de segmentation par lots. Pour plus d'informations sur l'utilisation des filtres, consultez la section Recommandations relatives au filtrage par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

jobInput

Le chemin Amazon S3 pour les données d'entrée utilisées pour générer la tâche de segment par lots.

Type : objet BatchSegmentJobInput

Obligatoire : oui

jobName

Nom de la tâche de segment par lots à créer.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

jobOutput

Le chemin Amazon S3 pour le compartiment dans lequel la sortie de la tâche sera stockée.

Type : objet BatchSegmentJobOutput

Obligatoire : oui

numResults

Le nombre d'utilisateurs prévus générés par la tâche de segmentation par lots pour chaque ligne de données d'entrée. Le nombre maximum d'utilisateurs par segment est de 5 millions.

Type : Integer

Obligatoire : non

roleArn

L'ARN du rôle Amazon Identity and Access Management qui est autorisé à lire et à écrire dans vos compartiments Amazon S3 d'entrée et de sortie respectivement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : oui

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de solution que vous souhaitez que la tâche de segment par lots utilise pour générer des segments par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Liste des balises à appliquer à la tâche de segmentation par lots.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "batchSegmentJobArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchSegmentJobArn

L'ARN de la tâche de segment par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateCampaign

Service : Amazon Personalize

> **Important**
>
> Vous engagez des frais de campagne pendant qu'elle est active. Pour éviter des coûts inutiles, veillez à supprimer la campagne lorsque vous aurez terminé. Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification Amazon Personalize.

Crée une campagne qui déploie une version de la solution. Lorsqu'un client appelle les GetPersonalizedRanking API GetRecommendations et, une campagne est spécifiée dans la demande.

TPS provisionné minimum et Auto-Scaling

> **Important**
>
> Un montant élevé `minProvisionedTPS` augmentera vos coûts. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre utilisation à l'aide CloudWatch des statistiques Amazon et `minProvisionedTPS` augmentez-la si nécessaire.

Lorsque vous créez une campagne Amazon Personalize, vous pouvez spécifier le nombre minimum de transactions provisionnées par seconde (`minProvisionedTPS`) pour la campagne. Il s'agit du débit de transaction de base pour la campagne provisionnée par Amazon Personalize. Il fixe les frais de facturation minimaux pour la campagne pendant qu'elle est active. Une transaction est une seule `GetRecommendations` ou une `GetPersonalizedRanking` demande. La valeur par défaut `minProvisionedTPS` est 1.

Si votre TPS augmente au-delà de`minProvisionedTPS`, Amazon Personalize redimensionne automatiquement la capacité provisionnée vers le haut et vers le bas, mais jamais en dessous. `minProvisionedTPS` Il y a un court délai pendant que la capacité est augmentée, ce qui peut entraîner une perte de transactions. Lorsque votre trafic diminue, la capacité revient au`minProvisionedTPS`.

Vous êtes facturé pour le TPS minimum provisionné ou, si vos demandes dépassent le TPS`minProvisionedTPS`, le TPS réel. Le TPS réel est le nombre total de demandes de recommandation que vous effectuez. Nous vous recommandons de commencer par une valeur faible`minProvisionedTPS`, de suivre votre consommation à l'aide CloudWatch des statistiques Amazon, puis d'augmenter cette `minProvisionedTPS` valeur si nécessaire.

Pour plus d'informations sur les coûts des campagnes, consultez la section Tarification Amazon Personalize.

Statut

Une campagne peut se présenter dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Pour connaître le statut de la campagne, appelez DescribeCampaign.

> **Note**
>
> Attendez la fin `status` de la campagne `ACTIVE` avant de demander des recommandations à la campagne.

API associées

- ListCampaigns
- DescribeCampaign
- UpdateCampaign
- DeleteCampaign

Syntaxe de la requête

```
{
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
```

```
      "rankingInfluence": {
         "string" : number
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
   "name": "string",
   "solutionVersionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

campaignConfig

Les détails de configuration d'une campagne.

Type : objet CampaignConfig

Obligatoire : non

minProvisionedTPS

Spécifie le nombre minimum de transactions provisionnées (recommandations) par seconde que Amazon Personalize prendra en charge. Un montant élevé `minProvisionedTPS` augmentera votre facture. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre utilisation à l'aide CloudWatch des statistiques Amazon et `minProvisionedTPS` augmentez-la si nécessaire.

Type : entier

Plage valide : Valeur minimum de 1.

Obligatoire : non

name

Un nom pour la nouvelle campagne. Le nom de la campagne doit être unique sur votre compte.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

solutionVersionArn

L'Amazon Resource Name (ARN) du modèle entraîné à déployer avec la campagne. Pour spécifier la dernière version de votre solution, spécifiez l'ARN de votre solution au `SolutionArn/$LATEST` format. Vous devez utiliser ce format si vous avez défini `syncWithLatestSolutionVersion` sur `True` dans le CampaignConfig.

Pour déployer un modèle qui n'est pas la dernière version de votre solution, spécifiez l'ARN de la version de la solution.

Pour plus d'informations sur les mises à jour automatiques des campagnes, consultez la section Activation des mises à jour automatiques des campagnes.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Liste des tags à appliquer à la campagne.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
```

```
   "campaignArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

campaignArn

Amazon Resource Name (ARN) de la campagne.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateDataDeletionJob

Service : Amazon Personalize

Crée une tâche par lots qui supprime toutes les références à des utilisateurs spécifiques d'un groupe de données Amazon Personalize par lots. Vous spécifiez les utilisateurs à supprimer dans un fichier CSV contenant des ID utilisateur dans un compartiment Amazon S3. Une fois une tâche terminée, Amazon Personalize ne s'entraîne plus sur les données des utilisateurs et ne prend plus en compte les utilisateurs lors de la génération de segments d'utilisateurs. Pour plus d'informations sur la création d'une tâche de suppression de données, consultez la section Suppression d'utilisateurs.

- Votre fichier d'entrée doit être un fichier CSV avec une seule colonne USER_ID répertoriant les ID des utilisateurs. Pour plus d'informations sur la préparation du fichier CSV, consultez Préparer votre fichier de suppression de données et le charger sur Amazon S3.
- Pour autoriser Amazon Personalize à accéder à votre fichier CSV d'entrée contenant les identifiants utilisateur, vous devez spécifier un rôle de service IAM autorisé à lire à partir de la source de données. Ce rôle a besoin d'`ListBucket`autorisations `GetObject` et d'autorisations pour le bucket et son contenu. Ces autorisations sont identiques à celles relatives à l'importation de données. Pour plus d'informations sur l'octroi de l'accès à votre compartiment Amazon S3, consultez la section Donner à Amazon un accès personnalisé aux ressources Amazon S3.

Une fois que vous avez créé une tâche, la suppression de toutes les références aux utilisateurs des jeux de données et des modèles peut prendre jusqu'à un jour. Jusqu'à la fin de la tâche, Amazon Personalize continue d'utiliser les données lors de la formation. Et si vous utilisez une recette de segmentation des utilisateurs, les utilisateurs peuvent apparaître dans les segments d'utilisateurs.

Statut

Une tâche de suppression de données peut avoir l'un des statuts suivants :

- EN ATTENTE > EN COURS > TERMINÉ -OU- ÉCHEC

Pour connaître l'état de la tâche de suppression de données, appelez DescribeDataDeletionJob API operation et spécifiez l'Amazon Resource Name (ARN) de la tâche. Si le statut est FAILED, la réponse inclut une `failureReason` clé qui décrit pourquoi la tâche a échoué.

API associées

- ListDataDeletionJobs

- DescribeDataDeletionJob

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "dataSource": {
      "dataLocation": "string"
   },
   "jobName": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données contenant les jeux de données dont vous souhaitez supprimer des enregistrements.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

dataSource

Le compartiment Amazon S3 qui contient la liste des ID utilisateur des utilisateurs à supprimer.

Type : objet DataSource

Obligatoire : oui

jobName

Le nom de la tâche de suppression de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

roleArn

Le nom de ressource Amazon (ARN) du rôle IAM autorisé à lire depuis la source de données Amazon S3.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : oui

tags

Liste des balises à appliquer à la tâche de suppression des données.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "dataDeletionJobArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

dataDeletionJobArn

Le nom de ressource Amazon (ARN) de la tâche de suppression des données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateDataset

Service : Amazon Personalize

Crée un jeu de données vide et l'ajoute au groupe de jeux de données spécifié. Permet d'CreateDatasetImportJobimporter vos données d'entraînement dans un jeu de données.

Il existe 5 types de jeux de données :

- Interactions entre objets
- Éléments
- Utilisateurs
- Interactions entre les actions
- Actions

Chaque type de jeu de données a un schéma associé avec des types de champs obligatoires. Seul le jeu de données `Item interactions` est nécessaire pour entraîner un modèle (également appelé création d'une solution).

Un jeu de données peut avoir l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Pour obtenir l'état de l'ensemble de données, appelez DescribeDataset.

API associées

- CreateDatasetGroup
- ListDatasets
- DescribeDataset
- DeleteDataset

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "datasetType": "string",
```

```
   "name": "string",
   "schemaArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données auquel ajouter l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

datasetType

Type de jeu de données.

L'une des valeurs suivantes (sans distinction majuscules/minuscules) :

- Interactions
- Éléments
- Utilisateurs
- Actions
- Action_Interactions

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `^[A-Za-z_]+$`

Obligatoire : oui

name

Le nom du jeu de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

schemaArn

L'ARN du schéma à associer à l'ensemble de données. Le schéma définit les champs de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Liste des balises à appliquer à l'ensemble de données.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetArn": "string"
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetArn

L'ARN du jeu de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d’état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateDatasetExportJob

Service : Amazon Personalize

Crée une tâche qui exporte les données de votre ensemble de données vers un compartiment Amazon S3. Pour autoriser Amazon Personalize à exporter les données de formation, vous devez spécifier un rôle IAM lié à un service qui donne à Amazon Personalize `PutObject` des autorisations pour votre compartiment Amazon S3. Pour plus d'informations, consultez la section Exporter un ensemble de données dans le guide du développeur Amazon Personalize.

Statut

Une tâche d'exportation de jeu de données peut se présenter dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Pour connaître l'état de la tâche d'exportation DescribeDatasetExportJob, appelez et spécifiez le nom de ressource Amazon (ARN) de la tâche d'exportation de l'ensemble de données. L'exportation de l'ensemble de données est terminée lorsque le statut est indiqué comme ACTIF. Si l'état indique CREATE FAILED, la réponse inclut une `failureReason` clé qui décrit pourquoi la tâche a échoué.

Syntaxe de la requête

```
{
   "datasetArn": "string",
   "ingestionMode": "string",
   "jobName": "string",
   "jobOutput": {
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données qui contient les données à exporter.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

ingestionMode

Les données à exporter, en fonction de la manière dont vous les avez importées. Vous pouvez choisir d'exporter uniquement les `BULK` données que vous avez importées à l'aide d'une tâche d'importation de jeux de données, uniquement les `PUT` données que vous avez importées de manière incrémentielle (à l'aide de la console PutUsers et PutItems des opérations), ou `ALL` pour les deux types. PutEvents La valeur par défaut est `PUT`.

Type : Chaîne

Valeurs valides : `BULK | PUT | ALL`

Obligatoire : non

jobName

Le nom de la tâche d'exportation de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

jobOutput

Le chemin d'accès au compartiment Amazon S3 dans lequel la sortie de la tâche est stockée.

Type : objet DatasetExportJobOutput

Obligatoire : oui

roleArn

Le nom de ressource Amazon (ARN) du rôle de service IAM autorisé à ajouter des données à votre compartiment Amazon S3 de sortie.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : oui

tags

Liste des balises à appliquer à la tâche d'exportation de l'ensemble de données.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetExportJobArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetExportJobArn

Le nom de ressource Amazon (ARN) de la tâche d'exportation de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateDatasetGroup

Service : Amazon Personalize

Crée un groupe de jeux de données vide. Un groupe de jeux de données est un conteneur pour les ressources Amazon Personalize. Un groupe de jeux de données peut contenir au maximum trois jeux de données, un pour chaque type de jeu de données :

- Interactions entre objets
- Éléments
- Utilisateurs
- Actions
- Interactions entre les actions

Un groupe de jeux de données peut être un groupe de jeux de données de domaine, dans lequel vous spécifiez un domaine et utilisez des ressources préconfigurées telles que des recommandations, ou un groupe de jeux de données personnalisé, dans lequel vous utilisez des ressources personnalisées, telles qu'une solution avec une version de solution, que vous déployez avec une campagne. Si vous commencez avec un groupe de jeux de données de domaine, vous pouvez toujours ajouter des ressources personnalisées telles que des solutions et des versions de solutions formées à l'aide de recettes pour des cas d'utilisation personnalisés et déployées avec des campagnes.

Un groupe de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- SUPPRIMER EN ATTENTE

Pour obtenir l'état du groupe de jeux de données, appelez DescribeDatasetGroup. Si l'état indique CREATE FAILED, la réponse inclut une `failureReason` clé qui décrit pourquoi la création a échoué.

> **Note**
>
> Vous devez attendre que le groupe `status` de jeux de données soit atteint `ACTIVE` avant d'ajouter un ensemble de données au groupe.

Vous pouvez spécifier une clé AWS Key Management Service (KMS) pour chiffrer les ensembles de données du groupe. Si vous spécifiez une clé KMS, vous devez également inclure un rôle Gestion des identités et des accès AWS (IAM) autorisé à accéder à cette clé.

API qui nécessitent un ARN de groupe d'ensembles de données dans la demande

- CreateDataset
- CreateEventTracker
- CreateSolution

API associées

- ListDatasetGroups
- DescribeDatasetGroup
- DeleteDatasetGroup

Syntaxe de la requête

```
{
   "domain": "string",
   "kmsKeyArn": "string",
   "name": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

domain

Le domaine du groupe de jeux de données. Spécifiez un domaine pour créer un groupe de jeux de données de domaine. Le domaine que vous spécifiez détermine les schémas par défaut pour

les ensembles de données et les cas d'utilisation disponibles pour les recommandateurs. Si vous ne spécifiez pas de domaine, vous créez un groupe de jeux de données personnalisé avec des versions de solution que vous déployez dans le cadre d'une campagne.

Type : Chaîne

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

kmsKeyArn

Le nom de ressource Amazon (ARN) d'une clé AWS Key Management Service (KMS) utilisée pour chiffrer les ensembles de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 2048.

Modèle : `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatoire : non

name

Le nom du nouveau groupe de jeux de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

roleArn

L'ARN du rôle Gestion des identités et des accès AWS (IAM) autorisé à accéder à la clé AWS Key Management Service (KMS). La fourniture d'un rôle IAM n'est valide que si vous spécifiez également une clé KMS.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

tags

Liste des balises à appliquer au groupe de jeux de données.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetGroupArn": "string",
   "domain": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetGroupArn

Le nom de ressource Amazon (ARN) du nouveau groupe de jeux de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

domain

Le domaine du nouveau groupe de jeux de données de domaine.

Type : Chaîne

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateDatasetImportJob

Service : Amazon Personalize

Crée une tâche qui importe les données de formation depuis votre source de données (un compartiment Amazon S3) vers un ensemble de données Amazon Personalize. Pour permettre à Amazon Personalize d'importer les données de formation, vous devez spécifier un rôle de service IAM autorisé à lire à partir de la source de données, car Amazon Personalize copie vos données et les traite en interne. Pour plus d'informations sur l'octroi de l'accès à votre compartiment Amazon S3, consultez la section Donner à Amazon un accès personnalisé aux ressources Amazon S3.

Si vous avez déjà créé un outil de recommandation ou déployé une version de solution personnalisée avec une campagne, la manière dont les nouveaux enregistrements groupés influencent les recommandations dépend du cas d'utilisation du domaine ou de la recette que vous utilisez. Pour plus d'informations, consultez la section Comment les nouvelles données influencent les recommandations en temps réel.

> **Important**
>
> Par défaut, une tâche d'importation de jeu de données remplace toutes les données existantes du jeu de données que vous avez importé en masse. Pour ajouter de nouveaux enregistrements sans remplacer les données existantes, spécifiez INCREMENTAL pour le mode d'importation de l' CreateDatasetImportJob opération.

Statut

Une tâche d'importation de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Pour connaître l'état de la tâche d'importation, appelez DescribeDatasetImportJob, en fournissant le nom de ressource Amazon (ARN) de la tâche d'importation de l'ensemble de données. L'importation d'ensembles de données est terminée quand le statut ACTIVE (ACTIF) apparaît. Si l'état indique CREATE FAILED, la réponse inclut une `failureReason` clé qui décrit pourquoi la tâche a échoué.

> **Note**
>
> L'importation nécessite un certain temps. Vous devez attendre que le statut soit ACTIF avant d'entraîner un modèle à l'aide du jeu de données.

API associées

- ListDatasetImportJobs
- DescribeDatasetImportJob

Syntaxe de la requête

```
{
   "datasetArn": "string",
   "dataSource": {
      "dataLocation": "string"
   },
   "importMode": "string",
   "jobName": "string",
   "publishAttributionMetricsToS3": boolean,
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

L'ARN du jeu de données qui reçoit les données importées.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

dataSource

Le compartiment Amazon S3 qui contient les données d'entraînement à importer.

Type : objet DataSource

Obligatoire : oui

importMode

Spécifiez comment ajouter les nouveaux enregistrements à un ensemble de données existant. Le mode d'importation par défaut est`FULL`. Si vous n'avez jamais importé d'enregistrements en masse dans le jeu de données auparavant, vous pouvez uniquement spécifier`FULL`.

- Spécifiez `FULL` de remplacer toutes les données groupées existantes dans votre jeu de données. Les données que vous avez importées individuellement ne sont pas remplacées.
- Spécifiez `INCREMENTAL` d'ajouter les nouveaux enregistrements aux données existantes de votre jeu de données. Amazon Personalize remplace tout enregistrement portant le même identifiant par le nouveau.

Type : Chaîne

Valeurs valides : `FULL | INCREMENTAL`

Obligatoire : non

jobName

Le nom de la tâche d'importation de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

publishAttributionMetricsToS3

Si vous avez créé une attribution de métrique, indiquez si vous souhaitez publier des métriques pour cette tâche d'importation sur Amazon S3

Type : booléen

Obligatoire : non

roleArn

L'ARN du rôle IAM autorisé à lire à partir de la source de données Amazon S3.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

tags

Liste des balises à appliquer à la tâche d'importation de l'ensemble de données.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetImportJobArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetImportJobArn

L'ARN de la tâche d'importation du jeu de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateEventTracker

Service : Amazon Personalize

Crée un outil de suivi des événements que vous utilisez lorsque vous ajoutez des données d'événements à un groupe d'ensembles de données spécifié à l'aide de l'PutEventsAPI.

> **Note**
>
> Un seul dispositif de suivi d'événement peut être associé à un groupe d'ensembles de données. Vous obtenez une erreur si vous appelez `CreateEventTracker` à l'aide du même groupe d'ensembles de données qu'un dispositif de suivi d'événement existant.

Lorsque vous créez un outil de suivi des événements, la réponse inclut un ID de suivi, que vous transmettez en paramètre lorsque vous utilisez l'PutEventsopération. Amazon Personalize ajoute ensuite les données d'événement à l'ensemble de données sur les interactions avec les articles du groupe de données que vous avez spécifié dans votre outil de suivi des événements.

Le système de suivi des événements peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Pour connaître l'état de l'outil de suivi des événements, appelez DescribeEventTracker.

> **Note**
>
> Le système de suivi des événements doit être à l'état ACTIF avant d'utiliser l'ID de suivi.

API associées

- ListEventTrackers
- DescribeEventTracker
- DeleteEventTracker

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "name": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données qui reçoit les données d'événement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

name

Le nom du système de suivi des événements.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

tags

Liste des balises à appliquer au système de suivi des événements.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "eventTrackerArn": "string",
   "trackingId": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

eventTrackerArn

L'ARN de l'outil de suivi des événements.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

trackingId

L'ID de l'outil de suivi des événements. Incluez cet identifiant dans les demandes adressées à l'PutEventsAPI.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript

- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateFilter

Service : Amazon Personalize

Crée un filtre de recommandation. Pour plus d'informations, consultez la section Filtrage des recommandations et des segments d'utilisateurs.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "filterExpression": "string",
   "name": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'ARN du groupe de jeux de données auquel le filtre appartiendra.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

filterExpression

L'expression de filtre définit les éléments qui sont inclus ou exclus des recommandations. L'expression du filtre doit suivre des règles de format spécifiques. Pour plus d'informations sur la structure et la syntaxe des expressions de filtre, voir Expressions de filtre.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 2500.

Obligatoire : oui

name

Le nom du filtre à créer.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

tags

Liste des balises à appliquer au filtre.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "filterArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

filterArn

L'ARN du nouveau filtre.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++

- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateMetricAttribution

Service : Amazon Personalize

Crée une attribution métrique. Une attribution métrique crée des rapports sur les données que vous importez dans Amazon Personalize. Selon la manière dont vous avez importé les données, vous pouvez consulter les rapports sur Amazon CloudWatch ou Amazon S3. Pour plus d'informations, voir Mesurer l'impact des recommandations.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "name": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données de destination pour l'attribution des métriques.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

metrics

Une liste d'attributs de métrique pour l'attribution de métrique. Chaque attribut de métrique spécifie un type d'événement à suivre et une fonction. Les fonctions disponibles sont `SUM()` ou`SAMPLECOUNT()`. Pour les fonctions SUM (), indiquez le type de jeu de données (Interactions ou Items) et la colonne à additionner en tant que paramètre. Par exemple SUM (Items.PRICE).

Type : tableau d'objets MetricAttribute

Membres du tableau : nombre maximum de 10 éléments.

Obligatoire : oui

metricsOutputConfig

Les détails de configuration de sortie pour l'attribution des métriques.

Type : objet MetricAttributionOutput

Obligatoire : oui

name

Nom pour l'attribution de la métrique.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "metricAttributionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metricAttributionArn

L'Amazon Resource Name (ARN) pour la nouvelle attribution de métrique.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateRecommender

Service : Amazon Personalize

Crée une recommandation avec la recette (un cas d'utilisation d'un groupe de jeux de données de domaine) que vous spécifiez. Vous créez des recommandations pour un groupe de jeux de données de domaine et vous spécifiez le nom de ressource Amazon (ARN) du recommandateur lorsque vous faites une GetRecommendations demande.

Nombre minimal de demandes de recommandation par seconde

> **Important**
>
> Un montant élevé `minRecommendationRequestsPerSecond` augmentera votre facture. Nous vous recommandons de commencer par 1 pour `minRecommendationRequestsPerSecond` (valeur par défaut). Suivez votre utilisation à l'aide CloudWatch des statistiques Amazon et `minRecommendationRequestsPerSecond` augmentez-la si nécessaire.

Lorsque vous créez un recommandateur, vous pouvez configurer le nombre minimum de demandes de recommandation par seconde du recommandateur. Le nombre minimal de demandes de recommandation par seconde (`minRecommendationRequestsPerSecond`) indique le débit de base des demandes de recommandation provisionné par Amazon Personalize. La valeur minimale par défaut RecommendationRequestsPerSecond est1. Une demande de recommandation est une `GetRecommendations` opération unique. Le débit des demandes est mesuré en demandes par seconde et Amazon Personalize utilise vos demandes par seconde pour calculer vos demandes par heure et le prix de l'utilisation de votre recommandation.

Si le nombre de demandes par seconde augmente au-delà`minRecommendationRequestsPerSecond`, Amazon Personalize redimensionne automatiquement la capacité provisionnée à la hausse et à la baisse, mais jamais en dessous. `minRecommendationRequestsPerSecond` Il y a un court délai pendant lequel la capacité est augmentée, ce qui peut entraîner une perte de demandes.

Votre facture correspond au plus élevé des deux montants suivants : le nombre minimum de demandes par heure (sur la base du minimumRecommendationRequestsPerSecond) ou le nombre réel de demandes. Le débit de requêtes réel utilisé est calculé comme la moyenne sur requests/second une période d'une heure. Nous vous recommandons de commencer par la valeur par

défautminRecommendationRequestsPerSecond, de suivre votre utilisation à l'aide CloudWatch des statistiques Amazon, puis de l'minRecommendationRequestsPerSecondaugmenter si nécessaire.

Statut

Un recommandateur peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- ARRÊTER EN ATTENTE > ARRÊTER EN COURS > INACTIF > DÉMARRER EN ATTENTE > DÉMARRER EN COURS > ACTIF
- DELETE PENDING > DELETE IN_PROGRESS

Pour obtenir le statut de recommandateur, appelez DescribeRecommender.

> **Note**
>
> Attendez que le `status` nom du recommandateur apparaisse `ACTIVE` avant de lui demander des recommandations.

API associées

- ListRecommenders
- DescribeRecommender
- UpdateRecommender
- DeleteRecommender

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "name": "string",
   "recipeArn": "string",
   "recommenderConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
```

```
      },
      "minRecommendationRequestsPerSecond": number,
      "trainingDataConfig": {
         "excludedDatasetColumns": {
            "string" : [ "string" ]
         },
         "includedDatasetColumns": {
            "string" : [ "string" ]
         }
      }
   },
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données du domaine de destination pour le recommandateur.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

name

Le nom du recommandateur.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

recipeArn

Le nom de ressource Amazon (ARN) de la recette que le recommandateur utilisera. Pour un recommandateur, une recette est un cas d'utilisation d'un groupe de jeux de données de domaine. Seuls les cas d'utilisation de groupes de jeux de données de domaine peuvent être utilisés pour créer une recommandation. Pour plus d'informations sur les cas d'utilisation, consultez la section Choix des cas d'utilisation recommandés.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

recommenderConfig

Les détails de configuration du recommandateur.

Type : objet RecommenderConfig

Obligatoire : non

tags

Une liste de balises à appliquer au recommandateur.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "recommenderArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recommenderArn

Le nom de ressource Amazon (ARN) du recommandateur.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateSchema

Service : Amazon Personalize

Crée un schéma Amazon Personalize à partir de la chaîne de schémas spécifiée. Le schéma que vous créez doit être au format Avro JSON.

Amazon Personalize reconnaît trois variantes de schéma. Chaque schéma est associé à un type de jeu de données et possède un ensemble de champs et de mots-clés obligatoires. Si vous créez un schéma pour un jeu de données dans un groupe de jeux de données Domaine, vous devez indiquer le domaine du groupe de jeux de données Domaine. Vous spécifiez un schéma lorsque vous appelez CreateDataset.

Pour plus d'informations sur les schémas, veuillez consulter la rubrique Jeux de données et schémas.

API associées

- ListSchemas
- DescribeSchema
- DeleteSchema

Syntaxe de la requête

```
{
   "domain": "string",
   "name": "string",
   "schema": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

domain

Le domaine du schéma. Si vous créez un schéma pour un jeu de données dans un groupe de jeux de données de domaine, spécifiez le domaine que vous avez choisi lors de la création du groupe de jeux de données de domaine.

Type : Chaîne

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

name

Nom du schéma.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

schema

Schéma au format Avro JSON.

Type : Chaîne

Contraintes de longueur : longueur maximale de 20 000.

Obligatoire : oui

Syntaxe de la réponse

```
{
   "schemaArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

schemaArn

Le nom de ressource Amazon (ARN) du schéma créé.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateSolution

Service : Amazon Personalize

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent la formation automatique. Avec la formation automatique, vous devez payer des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver la formation automatique. Pour plus d'informations sur les coûts de formation, consultez la section Tarification Amazon Personalize.

Crée la configuration pour entraîner un modèle (création d'une version de solution). Cette configuration inclut la recette à utiliser pour l'apprentissage des modèles et la configuration d'entraînement facultative, comme les colonnes à utiliser pour l'entraînement et les paramètres de transformation des fonctionnalités. Pour plus d'informations sur la configuration d'une solution, consultez la section Création et configuration d'une solution.

Par défaut, les nouvelles solutions utilisent la formation automatique pour créer des versions de solution tous les 7 jours. Vous pouvez modifier la fréquence d'entraînement. La création automatique de la version de la solution commence dans l'heure qui suit l'activation de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore la première formation automatique. Pour plus d'informations, consultez la section Configuration de l'entraînement automatique.

Pour désactiver l'entraînement automatique, réglez sur `performAutoTraining` False. Si vous désactivez la formation automatique, vous devez créer manuellement une version de la solution en appelant l'CreateSolutionVersionopération.

Une fois la formation commencée, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution grâce à l'opération ListSolutionVersions d'API. Pour obtenir son statut, utilisez le DescribeSolutionVersion.

Une fois la formation terminée, vous pouvez évaluer la précision du modèle en appelant GetSolutionMetrics. Lorsque vous êtes satisfait de la version de la solution, vous la déployez à l'aide de CreateCampaign. La campagne fournit des recommandations à un client via l'GetRecommendationsAPI.

> **Note**
>
> Amazon Personalize ne prend pas en charge la configuration de l'optimisation des hyperparamètres de la solution `hpoObjective` pour le moment.

Statut

Une solution peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Pour connaître l'état de la solution, appelez DescribeSolution. Si vous utilisez la formation manuelle, le statut doit être ACTIF avant d'appeler`CreateSolutionVersion`.

API associées

- UpdateSolution
- ListSolutions
- CreateSolutionVersion
- DescribeSolution
- DeleteSolution

- ListSolutionVersions
- DescribeSolutionVersion

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "eventType": "string",
   "name": "string",
   "performAutoML": boolean,
   "performAutoTraining": boolean,
   "performHPO": boolean,
   "performIncrementalUpdate": boolean,
   "recipeArn": "string",
```

```
   "solutionConfig": {
      "algorithmHyperParameters": {
         "string" : "string"
      },
      "autoMLConfig": {
         "metricName": "string",
         "recipeList": [ "string" ]
      },
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
      "eventsConfig": {
         "eventParametersList": [
            {
               "eventType": "string",
               "eventValueThreshold": number,
               "weight": number
            }
         ]
      },
      "eventValueThreshold": "string",
      "featureTransformationParameters": {
         "string" : "string"
      },
      "hpoConfig": {
         "algorithmHyperParameterRanges": {
            "categoricalHyperParameterRanges": [
               {
                  "name": "string",
                  "values": [ "string" ]
               }
            ],
            "continuousHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
               }
            ],
            "integerHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
```

```
                }
            ]
        },
        "hpoObjective": {
            "metricName": "string",
            "metricRegex": "string",
            "type": "string"
        },
        "hpoResourceConfig": {
            "maxNumberOfTrainingJobs": "string",
            "maxParallelTrainingJobs": "string"
        }
    },
    "optimizationObjective": {
        "itemAttribute": "string",
        "objectiveSensitivity": "string"
    },
    "trainingDataConfig": {
        "excludedDatasetColumns": {
            "string" : [ "string" ]
        },
        "includedDatasetColumns": {
            "string" : [ "string" ]
        }
    }
  },
  "tags": [
    {
        "tagKey": "string",
        "tagValue": "string"
    }
  ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données qui fournit les données d'entraînement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

eventType

Lorsque vous avez plusieurs types d'événements (à l'aide d'un champ de `EVENT_TYPE` schéma), ce paramètre indique quel type d'événement (par exemple, « clic » ou « J'aime ») est utilisé pour entraîner le modèle.

Si vous ne fournissez pas de`eventType`, Amazon Personalize utilisera toutes les interactions pour vous entraîner avec le même poids, quel que soit le type.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

name

Le nom de la solution.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : oui

performAutoML

> **Important**
>
> Nous vous déconseillons d'activer le machine learning. Adaptez plutôt les recettes Amazon Personalize disponibles à votre cas d'utilisation. Pour plus d'informations, consultez la section Choix d'une recette.

S'il faut effectuer un apprentissage automatique (AutoML). La valeur par défaut est `false`. Dans ce cas, vous devez spécifier`recipeArn`.

Lorsque cette option est définie sur`true`, Amazon Personalize analyse vos données d'entraînement et sélectionne la recette USER_PERSONALIZATION et les hyperparamètres optimaux. Dans ce cas, vous devez omettre`recipeArn`. Amazon Personalize détermine la recette optimale en exécutant des tests avec différentes valeurs pour les hyperparamètres. AutoML allonge le processus de formation par rapport à la sélection d'une recette spécifique.

Type : booléen

Obligatoire : non

performAutoTraining

Si la solution utilise la formation automatique pour créer de nouvelles versions de solution (modèles entraînés). Par défaut, `True` la solution crée automatiquement de nouvelles versions de solution tous les 7 jours. Vous pouvez modifier la fréquence d'entraînement en spécifiant un `schedulingExpression` dans `AutoTrainingConfig` la configuration de la solution. Pour plus d'informations sur l'entraînement automatique, voir Configuration de l'entraînement automatique.

La création automatique de la version de la solution commence dans l'heure qui suit l'activation de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore la première formation automatique.

Une fois la formation commencée, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution grâce à l'opération ListSolutionVersions d'API. Pour obtenir son statut, utilisez le DescribeSolutionVersion.

Type : booléen

Obligatoire : non

performHPO

S'il faut effectuer une optimisation des hyperparamètres (HPO) sur la recette spécifiée ou sélectionnée. La valeur par défaut est `false`.

Lorsque vous exécutez AutoML, ce paramètre est toujours `true` et vous ne devez pas le définir `false` sur.

Type : booléen

Obligatoire : non

performIncrementalUpdate

S'il faut effectuer des mises à jour d'entraînement incrémentielles sur votre modèle. Lorsqu'elle est activée, cela permet au modèle d'apprendre plus fréquemment à partir de nouvelles données sans nécessiter de reconversion complète, ce qui permet une personnalisation en temps quasi réel. Ce paramètre n'est pris en charge que pour les solutions qui utilisent la recette de similarité sémantique.

Type : booléen

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette à utiliser pour la formation des modèles. Ceci est obligatoire lorsque la valeur `performAutoML` est fausse. Pour plus d'informations sur les différentes recettes Amazon Personalize et leurs ARN, consultez la section Choix d'une recette.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionConfig

Les propriétés de configuration de la solution. Lorsqu'il `performAutoML` est défini sur true, Amazon Personalize évalue uniquement la `autoMLConfig` section de la configuration de la solution.

> **Note**
>
> Pour le moment, Amazon Personalize ne prend pas en charge la configuration de `hpoObjective`.

Type : objet SolutionConfig

Obligatoire : non

tags

Liste des balises à appliquer à la solution.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

Syntaxe de la réponse

```
{
   "solutionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

solutionArn

L'ARN de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d’état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

CreateSolutionVersion

Service : Amazon Personalize

Entraîne ou réentraîne une solution active dans un groupe de jeux de données personnalisé. Une solution est créée à l'aide de CreateSolution cette opération et doit être à l'état ACTIF avant d'appeler`CreateSolutionVersion`. Une nouvelle version de la solution est créée chaque fois que vous appelez cette opération.

Statut

Une version de solution peut se trouver dans l'un des états suivants :

- CRÉER EN ATTENTE
- CRÉER EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION
- CRÉER UN ARRÊT
- CRÉATION ARRÊTÉE

Pour connaître l'état de la version, appelez DescribeSolutionVersion. Attendez que le statut indique ACTIF avant d'appeler`CreateCampaign`.

Si l'état indique CREATE FAILED, la réponse inclut une `failureReason` clé qui décrit pourquoi la tâche a échoué.

API associées

- ListSolutionVersions
- DescribeSolutionVersion
- ListSolutions
- CreateSolution
- DescribeSolution
- DeleteSolution

Syntaxe de la requête

```
{
```

```
   "name": "string",
   "solutionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ],
   "trainingMode": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

name

Le nom de la version de la solution.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

solutionArn

Le nom de ressource Amazon (ARN) de la solution contenant les informations de configuration de la formation.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Liste des balises à appliquer à la version de la solution.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : non

trainingMode

Étendue de la formation à effectuer lors de la création de la version de la solution. La valeur par défaut est FULL. Cela crée un tout nouveau modèle basé sur l'intégralité des données d'entraînement provenant des ensembles de données de votre groupe de jeux de données.

Si vous utilisez User-Personalization, vous pouvez spécifier un mode d'entraînement deUPDATE. Cela met à jour le modèle afin de prendre en compte de nouveaux éléments pour les recommandations. Il ne s'agit pas d'une reconversion complète. Vous devriez tout de même effectuer une reconversion complète chaque semaine. Si vous le spécifiezUPDATE, Amazon Personalize arrêtera les mises à jour automatiques pour la version de la solution. Pour reprendre les mises à jour, créez une nouvelle solution avec le mode formation réglé sur FULL et déployez-la dans une campagne. Pour plus d'informations sur les mises à jour automatiques, voir Mises à jour automatiques.

L'UPDATEoption ne peut être utilisée que si vous disposez déjà d'une version de solution active créée à partir de la solution d'entrée à l'aide de l'FULLoption et que la solution d'entrée a été entraînée avec la User-Personalization recette ou l'ancienne HRNN-Coldstart recette.

Type : Chaîne

Valeurs valides : FULL | UPDATE | AUTOTRAIN

Obligatoire : non

Syntaxe de la réponse

```
{
   "solutionVersionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

solutionVersionArn

L'ARN de la nouvelle version de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteCampaign

Service : Amazon Personalize

Supprime une campagne en supprimant le déploiement de la solution. La solution sur laquelle repose la campagne n'est pas supprimée et peut être redéployée si nécessaire. Une campagne supprimée ne peut plus être spécifiée dans une GetRecommendations demande. Pour plus d'informations sur la création de campagnes, consultez CreateCampaign.

Syntaxe de la requête

```
{
   "campaignArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Le nom de ressource Amazon (ARN) de la campagne à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteDataset

Service : Amazon Personalize

Supprime un ensemble de données. Vous ne pouvez pas supprimer un jeu de données s'il est associé `DatasetImportJob` ou s'`SolutionVersion`il est dans l'état CREATE PENDING ou EN COURS. Pour plus d'informations sur la suppression de jeux de données, voir Supprimer un ensemble de données.

Syntaxe de la requête

```
{
   "datasetArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteDatasetGroup

Service : Amazon Personalize

Supprime un groupe de jeux de données. Avant de supprimer un groupe de jeux de données, vous devez supprimer les éléments suivants :

- Tous les dispositifs de suivi d'événements associés.
- Toutes les solutions associées.
- Tous les ensembles de données du groupe d'ensembles de données.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'ARN du groupe de jeux de données à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteEventTracker

Service : Amazon Personalize

Supprime le suivi des événements. Ne supprime pas l'ensemble de données du groupe de jeux de données. Pour plus d'informations sur les outils de suivi des événements, consultez CreateEventTracker.

Syntaxe de la requête

```
{
   "eventTrackerArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

eventTrackerArn

Le nom de ressource Amazon (ARN) de l'outil de suivi des événements à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteFilter

Service : Amazon Personalize

Supprime un filtre.

Syntaxe de la requête

```
{
   "filterArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

filterArn

L'ARN du filtre à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteMetricAttribution

Service : Amazon Personalize

Supprime une attribution de métrique.

Syntaxe de la requête

```
{
   "metricAttributionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

metricAttributionArn

L'Amazon Resource Name (ARN) de l'attribution de la métrique.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteRecommender

Service : Amazon Personalize

Désactive et supprime un recommandateur. Un recommandateur supprimé ne peut plus être spécifié dans une GetRecommendations demande.

Syntaxe de la requête

```
{
   "recommenderArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recommenderArn

Le nom de ressource Amazon (ARN) du recommandateur à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteSchema

Service : Amazon Personalize

Supprime un schéma. Avant de supprimer un schéma, vous devez supprimer tous les jeux de données qui y font référence. Pour plus d'informations sur les schémas, consultez CreateSchema.

Syntaxe de la requête

```
{
   "schemaArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

schemaArn

Le nom de ressource Amazon (ARN) du schéma à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DeleteSolution

Service : Amazon Personalize

Supprime toutes les versions d'une solution et l'`Solution`objet lui-même. Avant de supprimer une solution, vous devez supprimer toutes les campagnes basées sur cette solution. Pour déterminer quelles campagnes utilisent la solution, appelez ListCampaigns et indiquez l'Amazon Resource Name (ARN) de la solution. Vous ne pouvez pas supprimer une solution si une solution associée `SolutionVersion` est dans l'état CRÉER EN ATTENTE ou EN COURS. Pour plus d'informations sur les solutions, consultez CreateSolution.

Syntaxe de la requête

```
{
   "solutionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

solutionArn

L'ARN de la solution à supprimer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeAlgorithm

Service : Amazon Personalize

Décrit l'algorithme donné.

Syntaxe de la requête

```
{
   "algorithmArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

algorithmArn

Le nom de ressource Amazon (ARN) de l'algorithme à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "algorithm": {
      "algorithmArn": "string",
      "algorithmImage": {
         "dockerURI": "string",
         "name": "string"
      },
      "creationDateTime": number,
      "defaultHyperParameterRanges": {
         "categoricalHyperParameterRanges": [
            {
               "isTunable": boolean,
               "name": "string",
```

```
                "values": [ "string" ]
            }
         ],
         "continuousHyperParameterRanges": [
            {
               "isTunable": boolean,
               "maxValue": number,
               "minValue": number,
               "name": "string"
            }
         ],
         "integerHyperParameterRanges": [
            {
               "isTunable": boolean,
               "maxValue": number,
               "minValue": number,
               "name": "string"
            }
         ]
      },
      "defaultHyperParameters": {
         "string" : "string"
      },
      "defaultResourceConfig": {
         "string" : "string"
      },
      "lastUpdatedDateTime": number,
      "name": "string",
      "roleArn": "string",
      "trainingInputMode": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

algorithm

Liste des propriétés de l'algorithme.

Type : objet Algorithm

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeBatchInferenceJob

Service : Amazon Personalize

Obtient les propriétés d'une tâche d'inférence par lots, notamment le nom, le nom de la ressource Amazon (ARN), l'état, les configurations d'entrée et de sortie et l'ARN de la version de la solution utilisée pour générer les recommandations.

Syntaxe de la requête

```
{
   "batchInferenceJobArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

batchInferenceJobArn

L'ARN de la tâche d'inférence par lots à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "batchInferenceJob": {
      "batchInferenceJobArn": "string",
      "batchInferenceJobConfig": {
         "itemExplorationConfig": {
            "string" : "string"
         },
         "rankingInfluence": {
            "string" : number
```

```
            }
        },
        "batchInferenceJobMode": "string",
        "creationDateTime": number,
        "failureReason": "string",
        "filterArn": "string",
        "jobInput": {
            "s3DataSource": {
                "kmsKeyArn": "string",
                "path": "string"
            }
        },
        "jobName": "string",
        "jobOutput": {
            "s3DataDestination": {
                "kmsKeyArn": "string",
                "path": "string"
            }
        },
        "lastUpdatedDateTime": number,
        "numResults": number,
        "roleArn": "string",
        "solutionVersionArn": "string",
        "status": "string",
        "themeGenerationConfig": {
            "fieldsForThemeGeneration": {
                "itemName": "string"
            }
        }
    }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchInferenceJob

Informations sur la tâche d'inférence par lots spécifiée.

Type : objet BatchInferenceJob

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeBatchSegmentJob

Service : Amazon Personalize

Obtient les propriétés d'une tâche de segment par lots, notamment le nom, le nom de la ressource Amazon (ARN), l'état, les configurations d'entrée et de sortie et l'ARN de la version de la solution utilisée pour générer des segments.

Syntaxe de la requête

```
{
   "batchSegmentJobArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

batchSegmentJobArn

L'ARN de la tâche de segment par lots à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "batchSegmentJob": {
      "batchSegmentJobArn": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "filterArn": "string",
      "jobInput": {
         "s3DataSource": {
            "kmsKeyArn": "string",
```

```
            "path": "string"
         }
      },
      "jobName": "string",
      "jobOutput": {
         "s3DataDestination": {
            "kmsKeyArn": "string",
            "path": "string"
         }
      },
      "lastUpdatedDateTime": number,
      "numResults": number,
      "roleArn": "string",
      "solutionVersionArn": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchSegmentJob

Informations sur la tâche de segment de lot spécifiée.

Type : objet BatchSegmentJob

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeCampaign

Service : Amazon Personalize

Décrit la campagne donnée, y compris son statut.

Une campagne peut se présenter dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Lorsque `status` c'est`CREATE FAILED`, la réponse inclut la `failureReason` clé, qui décrit pourquoi.

Pour plus d'informations sur les campagnes, consultez CreateCampaign.

Syntaxe de la requête

```
{
   "campaignArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Amazon Resource Name (ARN) de la campagne.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "campaign": {
```

```
      "campaignArn": "string",
      "campaignConfig": {
         "enableMetadataWithRecommendations": boolean,
         "itemExplorationConfig": {
            "string" : "string"
         },
         "rankingInfluence": {
            "string" : number
         },
         "syncWithLatestSolutionVersion": boolean
      },
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "latestCampaignUpdate": {
         "campaignConfig": {
            "enableMetadataWithRecommendations": boolean,
            "itemExplorationConfig": {
               "string" : "string"
            },
            "rankingInfluence": {
               "string" : number
            },
            "syncWithLatestSolutionVersion": boolean
         },
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "minProvisionedTPS": number,
         "solutionVersionArn": "string",
         "status": "string"
      },
      "minProvisionedTPS": number,
      "name": "string",
      "solutionVersionArn": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

campaign

> **Note**
>
> Le `latestCampaignUpdate` champ n'est renvoyé que lorsque la campagne a reçu au moins un `UpdateCampaign` appel.

Les propriétés de la campagne.

> **Note**
>
> Le `latestCampaignUpdate` champ n'est renvoyé que lorsque la campagne a reçu au moins un `UpdateCampaign` appel.

Type : objet Campaign

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++

- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeDataDeletionJob

Service : Amazon Personalize

Décrit la tâche de suppression de données créée par CreateDataDeletionJob, y compris son statut.

Syntaxe de la requête

```
{
   "dataDeletionJobArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

dataDeletionJobArn

Le nom de ressource Amazon (ARN) de la tâche de suppression des données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "dataDeletionJob": {
      "creationDateTime": number,
      "dataDeletionJobArn": "string",
      "datasetGroupArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "numDeleted": number,
      "roleArn": "string",
```

```
        "status": "string"
    }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

dataDeletionJob

Informations sur la tâche de suppression des données, y compris son statut.

Le statut est l'une des valeurs suivantes :

- PENDING
- EN_COURS
- TERMINÉ
- ÉCHEC

Type : objet DataDeletionJob

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeDataset

Service : Amazon Personalize

Décrit l'ensemble de données donné. Pour plus d'informations sur les ensembles de données, consultez CreateDataset.

Syntaxe de la requête

```
{
   "datasetArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "dataset": {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetGroupArn": "string",
      "datasetType": "string",
      "lastUpdatedDateTime": number,
      "latestDatasetUpdate": {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "schemaArn": "string",
```

```
            "status": "string"
        },
        "name": "string",
        "schemaArn": "string",
        "status": "string",
        "trackingId": "string"
    }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

dataset

Liste des propriétés de l'ensemble de données.

Type : objet Dataset

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4

- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeDatasetExportJob

Service : Amazon Personalize

Décrit la tâche d'exportation de l'ensemble de données créée par CreateDatasetExportJob, y compris l'état de la tâche d'exportation.

Syntaxe de la requête

```
{
   "datasetExportJobArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetExportJobArn

Le nom de ressource Amazon (ARN) de la tâche d'exportation de l'ensemble de données à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "datasetExportJob": {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetExportJobArn": "string",
      "failureReason": "string",
      "ingestionMode": "string",
      "jobName": "string",
      "jobOutput": {
         "s3DataDestination": {
```

```
            "kmsKeyArn": "string",
            "path": "string"
         }
      },
      "lastUpdatedDateTime": number,
      "roleArn": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetExportJob

Informations sur la tâche d'exportation de l'ensemble de données, y compris son statut.

Le statut est l'une des valeurs suivantes :

- CRÉER EN ATTENTE
- CRÉER EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : objet DatasetExportJob

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeDatasetGroup

Service : Amazon Personalize

Décrit le groupe de jeux de données donné. Pour plus d'informations sur les groupes de jeux de données, consultez CreateDatasetGroup.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "datasetGroup": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "domain": "string",
      "failureReason": "string",
      "kmsKeyArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "roleArn": "string",
      "status": "string"
   }
```

```
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetGroup

Liste des propriétés du groupe de jeux de données.

Type : objet DatasetGroup

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin

- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeDatasetImportJob

Service : Amazon Personalize

Décrit la tâche d'importation de l'ensemble de données créée par CreateDatasetImportJob, y compris l'état de la tâche d'importation.

Syntaxe de la requête

```
{
   "datasetImportJobArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetImportJobArn

Le nom de ressource Amazon (ARN) de la tâche d'importation de l'ensemble de données à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "datasetImportJob": {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetImportJobArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "importMode": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
```

```
      "publishAttributionMetricsToS3": boolean,
      "roleArn": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetImportJob

Informations sur la tâche d'importation de l'ensemble de données, y compris son statut.

Le statut est l'une des valeurs suivantes :

- CRÉER EN ATTENTE
- CRÉER EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : objet DatasetImportJob

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeEventTracker

Service : Amazon Personalize

Décrit un outil de suivi des événements. La réponse inclut la fin `trackingId` du dispositif `status` de suivi des événements. Pour plus d'informations sur les outils de suivi des événements, consultez CreateEventTracker.

Syntaxe de la requête

```
{
   "eventTrackerArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

eventTrackerArn

Le nom de ressource Amazon (ARN) de l'outil de suivi des événements à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "eventTracker": {
      "accountId": "string",
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "eventTrackerArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string",
      "trackingId": "string"
   }
```

```
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

eventTracker

Un objet qui décrit le suivi des événements.

Type : objet EventTracker

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin

- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeFeatureTransformation

Service : Amazon Personalize

Décrit la transformation de fonction donnée.

Syntaxe de la requête

```
{
   "featureTransformationArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

featureTransformationArn

Le nom de ressource Amazon (ARN) de la transformation de fonctionnalité à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "featureTransformation": {
      "creationDateTime": number,
      "defaultParameters": {
         "string" : "string"
      },
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

featureTransformation

Une liste des FeatureTransformation propriétés.

Type : objet FeatureTransformation

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3

- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeFilter

Service : Amazon Personalize

Décrit les propriétés d'un filtre.

Syntaxe de la requête

```
{
   "filterArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

filterArn

L'ARN du filtre à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "filter": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "filterArn": "string",
      "filterExpression": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

filter

Les détails du filtre.

Type : objet Filter

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3

- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeMetricAttribution

Service : Amazon Personalize

Décrit une attribution métrique.

Syntaxe de la requête

```
{
   "metricAttributionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

metricAttributionArn

L'Amazon Resource Name (ARN) de l'attribution de la métrique.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "metricAttribution": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "metricAttributionArn": "string",
      "metricsOutputConfig": {
         "roleArn": "string",
         "s3DataDestination": {
            "kmsKeyArn": "string",
            "path": "string"
         }
```

```
        },
        "name": "string",
        "status": "string"
    }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metricAttribution

Détails de l'attribution des métriques.

Type : objet MetricAttribution

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2

- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeRecipe

Service : Amazon Personalize

Décrit une recette.

Une recette contient trois éléments :

- Algorithme qui entraîne un modèle.
- Hyperparamètres qui régissent l'entraînement.
- Informations sur la transformation des fonctionnalités permettant de modifier les données d'entrée avant l'entraînement.

Amazon Personalize propose un ensemble de recettes prédéfinies. Vous spécifiez une recette lorsque vous créez une solution à l'aide de l'CreateSolutionAPI. `CreateSolution`entraîne un modèle à l'aide de l'algorithme de la recette spécifiée et d'un ensemble de données d'apprentissage. La solution, lorsqu'elle est déployée sous forme de campagne, peut fournir des recommandations à l'aide de l'GetRecommendationsAPI.

Syntaxe de la requête

```
{
   "recipeArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recipeArn

Le nom de ressource Amazon (ARN) de la recette à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "recipe": {
      "algorithmArn": "string",
      "creationDateTime": number,
      "description": "string",
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "recipeType": "string",
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recipe

Un objet qui décrit la recette.

Type : objet Recipe

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeRecommender

Service : Amazon Personalize

Décrit le recommandateur donné, y compris son statut.

Un recommandateur peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- ARRÊTER EN ATTENTE > ARRÊTER EN COURS > INACTIF > DÉMARRER EN ATTENTE > DÉMARRER EN COURS > ACTIF
- DELETE PENDING > DELETE IN_PROGRESS

Lorsque `status` c'est`CREATE FAILED`, la réponse inclut la `failureReason` clé, qui décrit pourquoi.

La `modelMetrics` clé est nulle lorsque le recommandateur est créé ou supprimé.

Pour plus d'informations sur les recommandateurs, consultez CreateRecommender.

Syntaxe de la requête

```
{
   "recommenderArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recommenderArn

Le nom de ressource Amazon (ARN) du recommandateur à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "recommender": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "latestRecommenderUpdate": {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "recommenderConfig": {
            "enableMetadataWithRecommendations": boolean,
            "itemExplorationConfig": {
               "string" : "string"
            },
            "minRecommendationRequestsPerSecond": number,
            "trainingDataConfig": {
               "excludedDatasetColumns": {
                  "string" : [ "string" ]
               },
               "includedDatasetColumns": {
                  "string" : [ "string" ]
               }
            }
         },
         "status": "string"
      },
      "modelMetrics": {
         "string" : number
      },
      "name": "string",
      "recipeArn": "string",
      "recommenderArn": "string",
      "recommenderConfig": {
         "enableMetadataWithRecommendations": boolean,
         "itemExplorationConfig": {
            "string" : "string"
         },
         "minRecommendationRequestsPerSecond": number,
         "trainingDataConfig": {
            "excludedDatasetColumns": {
               "string" : [ "string" ]
```

```
            },
            "includedDatasetColumns": {
               "string" : [ "string" ]
            }
         }
      },
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recommender

Les propriétés du recommandeur.

Type : objet Recommender

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2

- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeSchema

Service : Amazon Personalize

Décrit un schéma. Pour plus d'informations sur les schémas, consultez CreateSchema.

Syntaxe de la requête

```
{
   "schemaArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

schemaArn

Le nom de ressource Amazon (ARN) du schéma à récupérer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "schema": {
      "creationDateTime": number,
      "domain": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "schema": "string",
      "schemaArn": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

schema

Le schéma demandé.

Type : objet DatasetSchema

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeSolution

Service : Amazon Personalize

Décrit une solution. Pour plus d'informations sur les solutions, consultez CreateSolution.

Syntaxe de la requête

```
{
   "solutionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

solutionArn

Le nom de ressource Amazon (ARN) de la solution à décrire.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "solution": {
      "autoMLResult": {
         "bestRecipeArn": "string"
      },
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "eventType": "string",
      "lastUpdatedDateTime": number,
      "latestSolutionUpdate": {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "performAutoTraining": boolean,
```

```
      "performIncrementalUpdate": boolean,
      "solutionUpdateConfig": {
         "autoTrainingConfig": {
            "schedulingExpression": "string"
         },
         "eventsConfig": {
            "eventParametersList": [
               {
                  "eventType": "string",
                  "eventValueThreshold": number,
                  "weight": number
               }
            ]
         }
      },
      "status": "string"
   },
   "latestSolutionVersion": {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string",
      "trainingMode": "string",
      "trainingType": "string"
   },
   "name": "string",
   "performAutoML": boolean,
   "performAutoTraining": boolean,
   "performHPO": boolean,
   "performIncrementalUpdate": boolean,
   "recipeArn": "string",
   "solutionArn": "string",
   "solutionConfig": {
      "algorithmHyperParameters": {
         "string" : "string"
      },
      "autoMLConfig": {
         "metricName": "string",
         "recipeList": [ "string" ]
      },
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
```

```
        "eventsConfig": {
           "eventParametersList": [
              {
                 "eventType": "string",
                 "eventValueThreshold": number,
                 "weight": number
              }
           ]
        },
        "eventValueThreshold": "string",
        "featureTransformationParameters": {
           "string" : "string"
        },
        "hpoConfig": {
           "algorithmHyperParameterRanges": {
              "categoricalHyperParameterRanges": [
                 {
                    "name": "string",
                    "values": [ "string" ]
                 }
              ],
              "continuousHyperParameterRanges": [
                 {
                    "maxValue": number,
                    "minValue": number,
                    "name": "string"
                 }
              ],
              "integerHyperParameterRanges": [
                 {
                    "maxValue": number,
                    "minValue": number,
                    "name": "string"
                 }
              ]
           },
           "hpoObjective": {
              "metricName": "string",
              "metricRegex": "string",
              "type": "string"
           },
           "hpoResourceConfig": {
              "maxNumberOfTrainingJobs": "string",
              "maxParallelTrainingJobs": "string"
```

```
            }
         },
         "optimizationObjective": {
            "itemAttribute": "string",
            "objectiveSensitivity": "string"
         },
         "trainingDataConfig": {
            "excludedDatasetColumns": {
               "string" : [ "string" ]
            },
            "includedDatasetColumns": {
               "string" : [ "string" ]
            }
         }
      },
      "status": "string"
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

solution

Un objet qui décrit la solution.

Type : objet Solution

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

DescribeSolutionVersion

Service : Amazon Personalize

Décrit une version spécifique d'une solution. Pour plus d'informations sur les solutions, voir CreateSolution

Syntaxe de la requête

```
{
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "solutionVersion": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "eventType": "string",
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "performAutoML": boolean,
      "performHPO": boolean,
      "performIncrementalUpdate": boolean,
      "recipeArn": "string",
      "solutionArn": "string",
```

```
   "solutionConfig": {
      "algorithmHyperParameters": {
         "string" : "string"
      },
      "autoMLConfig": {
         "metricName": "string",
         "recipeList": [ "string" ]
      },
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
      "eventsConfig": {
         "eventParametersList": [
            {
               "eventType": "string",
               "eventValueThreshold": number,
               "weight": number
            }
         ]
      },
      "eventValueThreshold": "string",
      "featureTransformationParameters": {
         "string" : "string"
      },
      "hpoConfig": {
         "algorithmHyperParameterRanges": {
            "categoricalHyperParameterRanges": [
               {
                  "name": "string",
                  "values": [ "string" ]
               }
            ],
            "continuousHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
               }
            ],
            "integerHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
```

```
                }
             ]
          },
          "hpoObjective": {
             "metricName": "string",
             "metricRegex": "string",
             "type": "string"
          },
          "hpoResourceConfig": {
             "maxNumberOfTrainingJobs": "string",
             "maxParallelTrainingJobs": "string"
          }
       },
       "optimizationObjective": {
          "itemAttribute": "string",
          "objectiveSensitivity": "string"
       },
       "trainingDataConfig": {
          "excludedDatasetColumns": {
             "string" : [ "string" ]
          },
          "includedDatasetColumns": {
             "string" : [ "string" ]
          }
       }
    },
    "solutionVersionArn": "string",
    "status": "string",
    "trainingHours": number,
    "trainingMode": "string",
    "trainingType": "string",
    "tunedHPOParams": {
       "algorithmHyperParameters": {
          "string" : "string"
       }
    }
 }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

solutionVersion

La version de la solution.

Type : objet SolutionVersion

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

GetSolutionMetrics

Service : Amazon Personalize

Obtient les métriques pour la version de solution spécifiée.

Syntaxe de la requête

```
{
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution pour laquelle vous souhaitez obtenir des métriques.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "metrics": {
      "string" : number
   },
   "solutionVersionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metrics

Les métriques de la version de la solution. Pour plus d'informations, consultez la section Évaluation d'une version de solution à l'aide de métriques.

Type : mappage chaîne à double

Entrées sur la carte : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 256.

solutionVersionArn

ARN de version de solution identique à celui spécifié dans la demande.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListBatchInferenceJobs

Service : Amazon Personalize

Obtient la liste des tâches d'inférence par lots qui ont été effectuées à partir d'une version de solution.

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de résultats de tâches d'inférence par lots à renvoyer sur chaque page. La valeur par défaut est 100.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Le jeton permettant de demander la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution à partir de laquelle les tâches d'inférence par lots ont été créées.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

Syntaxe de la réponse

```
{
   "batchInferenceJobs": [
      {
         "batchInferenceJobArn": "string",
         "batchInferenceJobMode": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchInferenceJobs

Une liste contenant des informations sur chaque tâche renvoyée.

Type : tableau d'objets BatchInferenceJobSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Le jeton à utiliser pour récupérer la page de résultats suivante. La valeur correspond au `null` moment où il n'y a plus de résultats à renvoyer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListBatchSegmentJobs

Service : Amazon Personalize

Obtient la liste des tâches de segmentation par lots qui ont été effectuées à partir d'une version de solution que vous spécifiez.

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de résultats de tâches segmentées par lots à renvoyer sur chaque page. La valeur par défaut est 100.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Le jeton permettant de demander la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de solution que les tâches de segment par lots ont utilisée pour générer des segments par lots.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

Syntaxe de la réponse

```
{
   "batchSegmentJobs": [
      {
         "batchSegmentJobArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

batchSegmentJobs

Une liste contenant des informations sur chaque tâche renvoyée.

Type : tableau d'objets BatchSegmentJobSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Le jeton à utiliser pour récupérer la page de résultats suivante. La valeur correspond au `null` moment où il n'y a plus de résultats à renvoyer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListCampaigns

Service : Amazon Personalize

Renvoie la liste des campagnes qui utilisent la solution donnée. Lorsqu'aucune solution n'est spécifiée, toutes les campagnes associées au compte sont répertoriées. La réponse fournit les propriétés de chaque campagne, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les campagnes, consultez CreateCampaign.

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de campagnes à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à ListCampaigns pour obtenir la prochaine série de campagnes (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

solutionArn

Le nom de ressource Amazon (ARN) de la solution pour laquelle répertorier les campagnes. Lorsqu'aucune solution n'est spécifiée, toutes les campagnes associées au compte sont répertoriées.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

Syntaxe de la réponse

```
{
   "campaigns": [
      {
         "campaignArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

campaigns

Une liste des campagnes.

Type : tableau d'objets CampaignSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de campagnes (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListDataDeletionJobs

Service : Amazon Personalize

Renvoie une liste des tâches de suppression de données pour un groupe de jeux de données classées par date de création, la plus récente en premier. Lorsqu'aucun groupe d'ensembles de données n'est spécifié, toutes les tâches de suppression de données associées au compte sont répertoriées. La réponse fournit les propriétés de chaque tâche, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les tâches de suppression de données, consultez la section Suppression d'utilisateurs.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données pour lequel répertorier les tâches de suppression de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de tâches de suppression de données à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListDataDeletionJobs` pour obtenir la prochaine série de tâches (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "dataDeletionJobs": [
      {
         "creationDateTime": number,
         "dataDeletionJobArn": "string",
         "datasetGroupArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

dataDeletionJobs

La liste des tâches de suppression de données.

Type : tableau d’objets DataDeletionJobSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de tâches de suppression de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3

- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListDatasetExportJobs

Service : Amazon Personalize

Renvoie une liste des tâches d'exportation d'ensembles de données qui utilisent l'ensemble de données donné. Lorsqu'aucun jeu de données n'est spécifié, toutes les tâches d'exportation d'ensemble de données associées au compte sont répertoriées. La réponse fournit les propriétés de chaque tâche d'exportation de jeu de données, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les tâches d'exportation de jeux de données, consultez CreateDatasetExportJob. Pour plus d'informations sur les ensembles de données, consultez CreateDataset.

Syntaxe de la requête

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) de l'ensemble de données pour lequel répertorier les tâches d'exportation de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de tâches d'exportation de jeux de données à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListDatasetExportJobs` pour obtenir la prochaine série de tâches d'exportation de jeux de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetExportJobs": [
      {
         "creationDateTime": number,
         "datasetExportJobArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetExportJobs

La liste des tâches d'exportation de jeux de données.

Type : tableau d'objets DatasetExportJobSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de tâches d'exportation de jeux de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3

- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListDatasetGroups

Service : Amazon Personalize

Renvoie une liste de groupes de jeux de données. La réponse fournit les propriétés de chaque groupe de jeux de données, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les groupes de jeux de données, consultez CreateDatasetGroup.

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de groupes de jeux de données à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListDatasetGroups` pour obtenir le prochain ensemble de groupes de jeux de données (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
```

```
   "datasetGroups": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "domain": "string",
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetGroups

La liste de vos groupes de jeux de données.

Type : tableau d'objets DatasetGroupSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir le prochain ensemble de groupes de jeux de données (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListDatasetImportJobs

Service : Amazon Personalize

Renvoie une liste des tâches d'importation d'ensembles de données qui utilisent l'ensemble de données donné. Lorsqu'aucun jeu de données n'est spécifié, toutes les tâches d'importation de jeux de données associées au compte sont répertoriées. La réponse fournit les propriétés de chaque tâche d'importation de jeux de données, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les tâches d'importation de jeux de données, consultez CreateDatasetImportJob. Pour plus d'informations sur les ensembles de données, consultez CreateDataset.

Syntaxe de la requête

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) de l'ensemble de données pour lequel répertorier les tâches d'importation de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de tâches d'importation de jeux de données à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListDatasetImportJobs` pour obtenir la prochaine série de tâches d'importation de jeux de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasetImportJobs": [
      {
         "creationDateTime": number,
         "datasetImportJobArn": "string",
         "failureReason": "string",
         "importMode": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetImportJobs

La liste des tâches d'importation de jeux de données.

Type : tableau d’objets DatasetImportJobSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de tâches d'importation de jeux de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3

- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListDatasets

Service : Amazon Personalize

Renvoie la liste des ensembles de données contenus dans le groupe de jeux de données donné. La réponse fournit les propriétés de chaque ensemble de données, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les ensembles de données, consultez CreateDataset.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données qui contient les jeux de données à répertorier.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de jeux de données à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListDatasets` pour obtenir la prochaine série de tâches d'importation de jeux de données (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "datasets": [
      {
         "creationDateTime": number,
         "datasetArn": "string",
         "datasetType": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasets

Tableau d’objets `Dataset`. Chaque objet fournit des informations de métadonnées.

Type : tableau d’objets DatasetSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir le prochain ensemble de jeux de données (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListEventTrackers

Service : Amazon Personalize

Renvoie la liste des outils de suivi des événements associés au compte. La réponse fournit les propriétés de chaque outil de suivi des événements, notamment le nom de la ressource Amazon (ARN) et l'ID de suivi. Pour plus d'informations sur les outils de suivi des événements, consultez CreateEventTracker.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'ARN d'un groupe de jeux de données utilisé pour filtrer la réponse.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de suiveurs d'événements à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListEventTrackers` pour obtenir le prochain ensemble de suiveurs d'événements (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "eventTrackers": [
      {
         "creationDateTime": number,
         "eventTrackerArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

eventTrackers

Une liste de suiveurs d'événements.

Type : tableau d'objets EventTrackerSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de suiveurs d'événements (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListFilters

Service : Amazon Personalize

Répertorie tous les filtres appartenant à un groupe de jeux de données donné.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'ARN du groupe de jeux de données qui contient les filtres.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de filtres à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListFilters` pour obtenir le prochain ensemble de filtres (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "Filters": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "failureReason": "string",
         "filterArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

Filters

Liste des filtres renvoyés.

Type : tableau d'objets FilterSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Un jeton permettant d'obtenir la prochaine série de filtres (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListMetricAttributionMetrics

Service : Amazon Personalize

Répertorie les mesures pour l'attribution des métriques.

Syntaxe de la requête

```
{
   "maxResults": number,
   "metricAttributionArn": "string",
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de mesures à renvoyer sur une page de résultats.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

metricAttributionArn

Le nom de ressource Amazon (ARN) de l'attribution de métrique pour laquelle vous souhaitez récupérer les attributs.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

nextToken

Spécifiez le jeton de pagination d'une demande précédente pour récupérer la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metrics

Les mesures pour l'attribution de métrique spécifiée.

Type : tableau d'objets MetricAttribute

Membres du tableau : nombre maximum de 10 éléments.

nextToken

Spécifiez le jeton de pagination d'une `ListMetricAttributionMetricsResponse` demande précédente pour récupérer la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListMetricAttributions

Service : Amazon Personalize

Répertorie les attributions des métriques.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le groupe de données des attributions de métriques Amazon Resource Name (ARN).

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum d'attributions de métriques à renvoyer sur une page de résultats.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Spécifiez le jeton de pagination d'une demande précédente pour récupérer la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "metricAttributions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "metricAttributionArn": "string",
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metricAttributions

La liste des attributions de métriques.

Type : tableau d'objets MetricAttributionSummary

Membres du tableau : nombre maximum de 100 éléments.

nextToken

Spécifiez le jeton de pagination d'une demande précédente pour récupérer la page de résultats suivante.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListRecipes

Service : Amazon Personalize

Renvoie la liste des recettes disponibles. La réponse fournit les propriétés de chaque recette, y compris le nom de ressource Amazon (ARN) de la recette.

Syntaxe de la requête

```
{
   "domain": "string",
   "maxResults": number,
   "nextToken": "string",
   "recipeProvider": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

domain

Filtre les recettes renvoyées par domaine pour un groupe de jeux de données de domaine. Seules les recettes (cas d'utilisation de groupes de jeux de données de domaine) pour ce domaine sont incluses dans la réponse. Si vous ne spécifiez pas de domaine, toutes les recettes sont renvoyées.

Type : Chaîne

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

maxResults

Le nombre maximum de recettes à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListRecipes` pour obtenir la prochaine série de recettes (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

recipeProvider

La valeur par défaut est `SERVICE`.

Type : Chaîne

Valeurs valides : `SERVICE`

Obligatoire : non

Syntaxe de la réponse

```
{
   "nextToken": "string",
   "recipes": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "status": "string"
      }
   ]
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton pour obtenir la prochaine série de recettes.

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

recipes

La liste des recettes disponibles.

Type : tableau d'objets RecipeSummary

Membres du tableau : nombre maximum de 100 éléments.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++

- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListRecommenders

Service : Amazon Personalize

Renvoie une liste de recommandations dans un groupe de jeux de données de domaine donné. Lorsqu'aucun groupe de jeux de données de domaine n'est spécifié, tous les recommandateurs associés au compte sont répertoriés. La réponse fournit les propriétés de chaque recommandateur, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les recommandateurs, consultez CreateRecommender.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de jeux de données de domaine pour lequel répertorier les personnes recommandées. Lorsqu'aucun groupe de jeux de données de domaine n'est spécifié, tous les recommandateurs associés au compte sont répertoriés.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de recommandateurs à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListRecommenders` pour obtenir le prochain ensemble de recommandations (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "nextToken": "string",
   "recommenders": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "recommenderArn": "string",
         "recommenderConfig": {
            "enableMetadataWithRecommendations": boolean,
            "itemExplorationConfig": {
               "string" : "string"
            },
            "minRecommendationRequestsPerSecond": number,
            "trainingDataConfig": {
               "excludedDatasetColumns": {
                  "string" : [ "string" ]
               },
               "includedDatasetColumns": {
                  "string" : [ "string" ]
               }
            }
         },
         "status": "string"
      }
   ]
```

```
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton permettant d'obtenir le prochain groupe de recommandations (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

recommenders

Une liste des recommandataires.

Type : tableau d’objets RecommenderSummary

Membres du tableau : nombre maximum de 100 éléments.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListSchemas

Service : Amazon Personalize

Renvoie la liste des schémas associés au compte. La réponse fournit les propriétés de chaque schéma, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les schémas, consultez CreateSchema.

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de schémas à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListSchemas` pour obtenir le prochain ensemble de schémas (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
```

```
   "nextToken": "string",
   "schemas": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "schemaArn": "string"
      }
   ]
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton utilisé pour obtenir le prochain ensemble de schémas (s'ils existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

schemas

Une liste de schémas.

Type : tableau d'objets DatasetSchemaSummary

Membres du tableau : nombre maximum de 100 éléments.

Erreurs

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListSolutions

Service : Amazon Personalize

Renvoie une liste de solutions dans un groupe de jeux de données donné. Lorsqu'aucun groupe de jeux de données n'est spécifié, toutes les solutions associées au compte sont répertoriées. La réponse fournit les propriétés de chaque solution, y compris l'Amazon Resource Name (ARN). Pour plus d'informations sur les solutions, consultez CreateSolution.

Syntaxe de la requête

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

maxResults

Le nombre maximum de solutions à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListSolutions` pour obtenir la prochaine série de solutions (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

Syntaxe de la réponse

```
{
   "nextToken": "string",
   "solutions": [
      {
         "creationDateTime": number,
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "solutionArn": "string",
         "status": "string"
      }
   ]
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton permettant d'obtenir la prochaine série de solutions (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

solutions

Liste des solutions actuelles.

Type : tableau d'objets SolutionSummary

Membres du tableau : nombre maximum de 100 éléments.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListSolutionVersions

Service : Amazon Personalize

Renvoie une liste des versions de solution pour la solution donnée. Lorsqu'aucune solution n'est spécifiée, toutes les versions de solution associées au compte sont répertoriées. La réponse fournit les propriétés de chaque version de solution, y compris l'Amazon Resource Name (ARN).

Syntaxe de la requête

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

maxResults

Le nombre maximum de versions de solution à renvoyer.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

nextToken

Un jeton renvoyé lors de l'appel précédent à `ListSolutionVersions` pour obtenir le prochain ensemble de versions de solution (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

Obligatoire : non

solutionArn

L'Amazon Resource Name (ARN) de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

Syntaxe de la réponse

```
{
   "nextToken": "string",
   "solutionVersions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string",
         "trainingMode": "string",
         "trainingType": "string"
      }
   ]
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton permettant d'obtenir le prochain ensemble de versions de la solution (si elles existent).

Type : Chaîne

Contraintes de longueur : longueur maximale de 1500.

Modèle : `\p{ASCII}{0,1500}`

solutionVersions

Liste des versions de solution décrivant les propriétés de la version.

Type : tableau d'objets SolutionVersionSummary

Membres du tableau : nombre maximum de 100 éléments.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

InvalidNextTokenException

Le jeton n'est pas valide.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

ListTagsForResource

Service : Amazon Personalize

Obtenez la liste des balises associées à une ressource.

Syntaxe de la requête

```
{
   "resourceArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

resourceArn

Nom de ressource Amazon (ARN) de la ressource.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

tags

Les balises de la ressource.

Type : tableau d'objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript

- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

StartRecommender

Service : Amazon Personalize

Démarre une recommandation qui est INACTIVE. Le lancement d'un recommandataire ne crée aucun nouveau modèle, mais relance la facturation et la reconversion automatique du recommandateur.

Syntaxe de la requête

```
{
   "recommenderArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recommenderArn

Le nom de ressource Amazon (ARN) de la personne qui recommande de commencer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "recommenderArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recommenderArn

Le nom de ressource Amazon (ARN) de la recommandation que vous avez lancée.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript

- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

StopRecommender

Service : Amazon Personalize

Arrête un recommandateur qui est ACTIF. L'arrêt d'un recommandataire interrompt la facturation et la reconversion automatique du recommandateur.

Syntaxe de la requête

```
{
   "recommenderArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recommenderArn

Le nom de ressource Amazon (ARN) de la personne à recommander à arrêter.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "recommenderArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recommenderArn

Le nom de ressource Amazon (ARN) de la recommandation que vous avez arrêtée.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)

- AWS SDK pour Ruby V3

StopSolutionVersionCreation

Service : Amazon Personalize

Arrête de créer une version de solution dont l'état est CREATE_PENDING ou CREATE IN_PROGRESS.

En fonction de l'état actuel de la version de la solution, l'état de la version de la solution change comme suit :

- CREATE_PENDING > CREATE_STOPPED

  or

- CREATE_IN_PROGRESS > CREATE_STOP > CREATE_STOPPED

Toutes les formations effectuées jusqu'à l'arrêt de la création de la version de la solution vous sont facturées. Vous ne pouvez pas reprendre la création d'une version de solution une fois celle-ci arrêtée.

Syntaxe de la requête

```
{
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution que vous souhaitez arrêter de créer.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

TagResource

Service : Amazon Personalize

Ajoutez une liste de balises à une ressource.

Syntaxe de la requête

```
{
   "resourceArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

resourceArn

Nom de ressource Amazon (ARN) de la ressource.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tags

Balises à appliquer à la ressource. Pour plus d'informations, consultez la section Marquage des ressources Amazon Personalize.

Type : tableau d’objets Tag

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Obligatoire : oui

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

TooManyTagsException

Vous avez dépassé le nombre maximum de balises que vous pouvez appliquer à cette ressource.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++

- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UntagResource

Service : Amazon Personalize

Supprime les balises spécifiées qui sont attachées à une ressource. Pour plus d'informations, consultez la section Supprimer des balises des ressources Amazon Personalize.

Syntaxe de la requête

```
{
   "resourceArn": "string",
   "tagKeys": [ "string" ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

resourceArn

Nom de ressource Amazon (ARN) de la ressource.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

tagKeys

Les clés des étiquettes à supprimer.

Type : tableau de chaînes

Membres du tableau : nombre minimum de 0 élément. Nombre maximum de 200 éléments.

Contraintes de longueur : longueur minimale de 1 caractère. Longueur maximale de 128.

Modèle : `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatoire : oui

Éléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

TooManyTagKeysException

La demande contient plus de clés de balise que ce qui peut être associé à une ressource (50 clés de balise par ressource).

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript

- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UpdateCampaign

Service : Amazon Personalize

Met à jour une campagne pour déployer une version de solution adaptée à une campagne existante, modifier celle de `minProvisionedTPS` votre campagne ou modifier la configuration de votre campagne. Par exemple, vous pouvez `enableMetadataWithRecommendations` définir la valeur true pour une campagne existante.

Pour mettre à jour une campagne afin qu'elle démarre automatiquement à l'aide de la dernière version de la solution, spécifiez les éléments suivants :

- Pour le `SolutionVersionArn` paramètre, spécifiez le nom de ressource Amazon (ARN) de votre solution au `SolutionArn/$LATEST` format.
- Dans le`campaignConfig`, réglez `syncWithLatestSolutionVersion` sur`true`.

Pour mettre à jour une campagne, son statut doit être ACTIF ou CREATE FAILED. Vérifiez l'état de la campagne à l'aide de l'DescribeCampaignopération.

> **Note**
>
> Vous pouvez toujours obtenir les recommandations d'une campagne pendant qu'une mise à jour est en cours. La campagne utilisera la version précédente de la solution et la configuration de la campagne pour générer des recommandations jusqu'à ce que le statut de la dernière mise à jour de la campagne soit atteint`Active`.

Pour plus d'informations sur la mise à jour d'une campagne, y compris des exemples de code, voir Mettre à jour une campagne. Pour plus d'informations sur les campagnes, consultez la section Création d'une campagne.

Syntaxe de la requête

```
{
   "campaignArn": "string",
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
```

```
      "rankingInfluence": {
         "string" : number
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
   "solutionVersionArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Amazon Resource Name (ARN) de la campagne.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

campaignConfig

Les détails de configuration d'une campagne.

Type : objet CampaignConfig

Obligatoire : non

minProvisionedTPS

Spécifie le nombre minimum de transactions provisionnées (recommandations) par seconde que Amazon Personalize prendra en charge. Un montant élevé `minProvisionedTPS` augmentera votre facture. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre utilisation à l'aide CloudWatch des statistiques Amazon et `minProvisionedTPS` augmentez-la si nécessaire.

Type : entier

Plage valide : Valeur minimum de 1.

Obligatoire : non

solutionVersionArn

L'Amazon Resource Name (ARN) d'un nouveau modèle à déployer. Pour spécifier la dernière version de votre solution, spécifiez l'ARN de votre solution au `SolutionArn/$LATEST` format. Vous devez utiliser ce format si vous avez défini `syncWithLatestSolutionVersion` sur `True` dans le CampaignConfig.

Pour déployer un modèle qui n'est pas la dernière version de votre solution, spécifiez l'ARN de la version de la solution.

Pour plus d'informations sur les mises à jour automatiques des campagnes, consultez la section Activation des mises à jour automatiques des campagnes.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

Syntaxe de la réponse

```
{
   "campaignArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

campaignArn

Le même ARN de campagne que celui indiqué dans la demande.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UpdateDataset

Service : Amazon Personalize

Mettez à jour un jeu de données pour remplacer son schéma par un nouveau schéma ou un schéma existant. Pour plus d'informations, consultez la section Remplacement du schéma d'un jeu de données.

Syntaxe de la requête

```
{
   "datasetArn": "string",
   "schemaArn": "string"
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données que vous souhaitez mettre à jour.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

schemaArn

L'Amazon Resource Name (ARN) du nouveau schéma que vous souhaitez utiliser.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
{
   "datasetArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données que vous avez mis à jour.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UpdateMetricAttribution

Service : Amazon Personalize

Met à jour une attribution de métrique.

Syntaxe de la requête

```
{
   "addMetrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricAttributionArn": "string",
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "removeMetrics": [ "string" ]
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

addMetrics

Ajoutez de nouveaux attributs de métrique à l'attribution de métrique.

Type : tableau d’objets MetricAttribute

Membres du tableau : nombre maximum de 10 éléments.

Obligatoire : non

metricAttributionArn

L'Amazon Resource Name (ARN) pour l'attribution de la métrique à mettre à jour.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

metricsOutputConfig

Une configuration de sortie pour l'attribution des métriques.

Type : objet MetricAttributionOutput

Obligatoire : non

removeMetrics

Supprimez les attributs de métrique de l'attribution de métrique.

Type : tableau de chaînes

Membres du tableau : nombre maximum de 10 éléments.

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
{
   "metricAttributionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

metricAttributionArn

L'Amazon Resource Name (ARN) pour l'attribution de métrique que vous avez mise à jour.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceAlreadyExistsException

La ressource spécifiée existe déjà.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2

- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UpdateRecommender

Service : Amazon Personalize

Met à jour le recommandateur pour modifier sa configuration. Si vous mettez à jour le module de recommandation pour modifier les colonnes utilisées lors de la formation, Amazon Personalize lance automatiquement un nouvel entraînement complet des modèles sur lesquels repose votre recommandation. Une fois la mise à jour terminée, vous pouvez toujours obtenir des recommandations de la part du recommandateur. Le recommandateur utilise la configuration précédente jusqu'à la fin de la mise à jour. Pour suivre l'état de cette mise à jour, utilisez les données `latestRecommenderUpdate` renvoyées dans l'DescribeRecommenderopération.

Syntaxe de la requête

```
{
   "recommenderArn": "string",
   "recommenderConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "minRecommendationRequestsPerSecond": number,
      "trainingDataConfig": {
         "excludedDatasetColumns": {
            "string" : [ "string" ]
         },
         "includedDatasetColumns": {
            "string" : [ "string" ]
         }
      }
   }
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

recommenderArn

Le nom de ressource Amazon (ARN) du recommandateur à modifier.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

recommenderConfig

Les détails de configuration du recommandateur.

Type : objet RecommenderConfig

Obligatoire : oui

Syntaxe de la réponse

```
{
   "recommenderArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

recommenderArn

Le même nom de ressource Amazon de recommandation (ARN) que celui indiqué dans la demande.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

UpdateSolution

Service : Amazon Personalize

Met à jour une solution Amazon Personalize afin d'utiliser une configuration d'entraînement automatique différente. Lorsque vous mettez à jour une solution, vous pouvez modifier si la solution utilise l'entraînement automatique, et vous pouvez modifier la fréquence de l'entraînement. Pour plus d'informations sur la mise à jour d'une solution, consultez la section Mise à jour d'une solution.

Une mise à jour de solution peut se présenter dans l'un des états suivants :

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Pour connaître l'état d'une mise à jour de solution, appelez l'opération DescribeSolution API et recherchez l'état dans le`latestSolutionUpdate`.

Syntaxe de la requête

```
{
   "performAutoTraining": boolean,
   "performIncrementalUpdate": boolean,
   "solutionArn": "string",
   "solutionUpdateConfig": {
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
      "eventsConfig": {
         "eventParametersList": [
            {
               "eventType": "string",
               "eventValueThreshold": number,
               "weight": number
            }
         ]
      }
   }
}
```

Paramètres de demande

Cette demande accepte les données suivantes au format JSON.

performAutoTraining

Si la solution utilise la formation automatique pour créer de nouvelles versions de solution (modèles entraînés). Vous pouvez modifier la fréquence d'entraînement en spécifiant un `schedulingExpression` dans `AutoTrainingConfig` la configuration de la solution.

Si vous activez la formation automatique, la première formation automatique commence dans l'heure qui suit la fin de la mise à jour de la solution. Si vous créez manuellement une version de solution dans l'heure qui suit, la solution ignore la première formation automatique. Pour plus d'informations sur l'entraînement automatique, consultez la section Configuration de l'entraînement automatique.

Une fois la formation commencée, vous pouvez obtenir le nom de ressource Amazon (ARN) de la version de la solution grâce à l'opération ListSolutionVersions d'API. Pour obtenir son statut, utilisez le DescribeSolutionVersion.

Type : booléen

Obligatoire : non

performIncrementalUpdate

S'il faut effectuer des mises à jour d'entraînement incrémentielles sur votre modèle. Lorsqu'elle est activée, cela permet au modèle d'apprendre plus fréquemment à partir de nouvelles données sans nécessiter de reconversion complète, ce qui permet une personnalisation en temps quasi réel. Ce paramètre n'est pris en charge que pour les solutions qui utilisent la recette de similarité sémantique.

Type : booléen

Obligatoire : non

solutionArn

Le nom de ressource Amazon (ARN) de la solution à mettre à jour.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

solutionUpdateConfig

Les nouveaux détails de configuration de la solution.

Type : objet SolutionUpdateConfig

Obligatoire : non

Syntaxe de la réponse

```
{
   "solutionArn": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

solutionArn

La même solution Amazon Resource Name (ARN) que celle indiquée dans la demande.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

LimitExceededException

La limite du nombre de requêtes par seconde a été dépassée.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 400

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d’état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

Amazon Personalize Events

Les actions suivantes sont prises en charge par Amazon Personalize Events :

- PutActionInteractions

- PutActions
- PutEvents
- PutItems
- PutUsers

PutActionInteractions

Service : Amazon Personalize Events

Enregistre les données relatives aux événements d'interaction entre actions. Un événement d'interaction entre une action est une interaction entre un utilisateur et une action. Par exemple, un utilisateur effectuant une action, comme s'inscrire à un programme d'adhésion ou télécharger votre application.

Pour plus d'informations sur l'enregistrement des interactions entre actions, voir Enregistrement des événements d'interaction entre actions. Pour plus d'informations sur les actions d'un jeu de données Actions, voir Ensemble de données Actions.

Syntaxe de la demande

```
POST /action-interactions HTTP/1.1
Content-type: application/json

{
   "actionInteractions": [
      {
         "actionId": "string",
         "eventId": "string",
         "eventType": "string",
         "impression": [ "string" ],
         "properties": "string",
         "recommendationId": "string",
         "sessionId": "string",
         "timestamp": number,
         "userId": "string"
      }
   ],
   "trackingId": "string"
}
```

Paramètres de demande URI

La demande n’utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

actionInteractions

Liste des événements d'interaction entre actions survenus au cours de la session.

Type : tableau d’objets ActionInteraction

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 10 éléments.

Obligatoire : oui

trackingId

L'identifiant de votre outil de suivi des événements et des interactions entre les actions. Lorsque vous créez un jeu de données Action Interactions, Amazon Personalize crée pour vous un outil de suivi des événements liés aux interactions entre actions. Pour plus d'informations, consultez la section ID du suivi des événements Action interaction.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d’état HTTP : 409

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

PutActions

Service : Amazon Personalize Events

Ajoute une ou plusieurs actions à un jeu de données Actions. Pour plus d'informations, voir Importer des actions individuellement.

Syntaxe de la demande

```
POST /actions HTTP/1.1
Content-type: application/json

{
   "actions": [
      {
         "actionId": "string",
         "properties": "string"
      }
   ],
   "datasetArn": "string"
}
```

Paramètres de demande URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

actions

Liste des données relatives aux actions.

Type : tableau d'objets Action

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 10 éléments.

Obligatoire : oui

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données Actions auquel vous ajoutez l'action ou les actions.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 409

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2

- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

PutEvents

Service : Amazon Personalize Events

Enregistre les données relatives aux événements d'interaction entre les objets. Pour plus d'informations, consultez la section Enregistrement des événements d'interaction entre les éléments.

> **Note**
>
> Si vous utilisez une AWS Lambda fonction pour appeler l' PutEvents opération, le rôle d'exécution de votre fonction doit être autorisé à effectuer l'`personalize:PutEvents`action avec le caractère générique `*` dans l'`Resource`élément.

Syntaxe de la demande

```
POST /events HTTP/1.1
Content-type: application/json

{
   "eventList": [
      {
         "eventId": "string",
         "eventType": "string",
         "eventValue": number,
         "impression": [ "string" ],
         "itemId": "string",
         "metricAttribution": {
            "eventAttributionSource": "string"
         },
         "properties": "string",
         "recommendationId": "string",
         "sentAt": number
      }
   ],
   "sessionId": "string",
   "trackingId": "string",
   "userId": "string"
}
```

Paramètres de demande URI

La demande n’utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

eventList

Liste des données relatives aux événements de la session.

Type : tableau d'objets Event

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 10 éléments.

Obligatoire : oui

sessionId

L'identifiant de session associé à la visite de l'utilisateur. Votre application génère le SessionID lorsqu'un utilisateur visite pour la première fois votre site Web ou utilise votre application. Amazon Personalize utilise le SessionID pour associer des événements à l'utilisateur avant qu'il ne se connecte. Pour plus d'informations, consultez la section Enregistrement des événements d'interaction entre les éléments.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

trackingId

L'identifiant de suivi de l'événement. L'ID est généré par un appel à l'CreateEventTrackerAPI.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

userId

L'utilisateur associé à l'événement.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

PutItems

Service : Amazon Personalize Events

Ajoute un ou plusieurs éléments à un jeu de données Items. Pour plus d'informations, consultez la section Importation d'articles individuellement.

Syntaxe de la demande

```
POST /items HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "items": [
      {
         "itemId": "string",
         "properties": "string"
      }
   ]
}
```

Paramètres de demande URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données Items auquel vous ajoutez l'article ou les articles.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

items

Liste des données relatives aux articles.

Type : tableau d'objets Item

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 10 éléments.

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 409

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

PutUsers

Service : Amazon Personalize Events

Ajoute un ou plusieurs utilisateurs à un ensemble de données Utilisateurs. Pour plus d'informations, voir Importer des utilisateurs individuellement.

Syntaxe de la demande

```
POST /users HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "users": [
      {
         "properties": "string",
         "userId": "string"
      }
   ]
}
```

Paramètres de demande URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

datasetArn

Le nom de ressource Amazon (ARN) du jeu de données Users auquel vous ajoutez l'utilisateur ou les utilisateurs.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

users

Liste des données des utilisateurs.

Type : tableau d'objets User

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 10 éléments.

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceInUseException

La ressource spécifiée est en cours d'utilisation.

Code d'état HTTP : 409

ResourceNotFoundException

Impossible de trouver la ressource spécifiée.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

Amazon Personalize Runtime

Les actions suivantes sont prises en charge par Amazon Personalize Runtime :

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

GetActionRecommendations

Service : Amazon Personalize Runtime

Renvoie une liste des actions recommandées, classées par ordre décroissant par score de prédiction. Utilisez l'`GetActionRecommendations`API si vous avez une campagne personnalisée qui déploie une version de solution entraînée à l'aide d'une recette PERSONALIZED_ACTIONS.

Pour plus d'informations sur les recettes PERSONALIZED_ACTIONS, consultez les recettes PERSONALIZED_ACTIONS. https://docs.aws.amazon.com/personalize/latest/dg/nexts-best-action-recipes.html Pour plus d'informations sur l'obtention de recommandations d'action, voir Obtenir des recommandations d'action.

Syntaxe de la demande

```
POST /action-recommendations HTTP/1.1
Content-type: application/json

{
   "campaignArn": "string",
   "filterArn": "string",
   "filterValues": {
      "string" : "string"
   },
   "numResults": number,
   "userId": "string"
}
```

Paramètres de demande URI

La demande n’utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Le nom de ressource Amazon (ARN) de la campagne à utiliser pour obtenir des recommandations d'action. Cette campagne doit déployer une version de solution entraînée à l'aide d'une recette PERSONALIZED_ACTIONS.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterArn

L'ARN du filtre à appliquer aux recommandations renvoyées. Pour de plus amples informations, veuillez consulter Filtrage des recommandations.

Lorsque vous utilisez ce paramètre, assurez-vous que la ressource de filtre l'est`ACTIVE`.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterValues

Les valeurs à utiliser lors du filtrage des recommandations. Pour chaque paramètre d'espace réservé dans votre expression de filtre, indiquez le nom du paramètre (dans les cas correspondants) en tant que clé et la ou les valeurs du filtre en tant que valeur correspondante. Séparez les valeurs multiples d'un paramètre par une virgule.

Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des actions, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des actions, vous pouvez omettre le`filter-values`. Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

Pour plus d'informations, consultez la section Filtrage des recommandations et des segments d'utilisateurs.

Type : mappage chaîne/chaîne

Entrées sur la carte : nombre maximum de 25 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 50.

Modèle de clé :`[A-Za-z0-9_]+`

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

numResults

Le nombre de résultats à renvoyer. La valeur par défaut est 5. Le maximum est de 100.

Type : Integer

Plage valide : Valeur minimum de 0.

Obligatoire : non

userId

L'ID utilisateur de l'utilisateur pour lequel fournir des recommandations d'action.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "actionList": [
      {
         "actionId": "string",
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

actionList

Liste de recommandations d'action triées par ordre décroissant par score de prédiction. Il peut y avoir un maximum de 100 actions dans la liste. Pour plus d'informations sur les scores d'action, consultez Comment fonctionne la notation des recommandations d'action.

Type : tableau d’objets PredictedAction

recommendationId

L'ID de la recommandation.

Type : Chaîne

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d’état HTTP : 400

ResourceNotFoundException

La ressource spécifiée n'existe pas.

Code d’état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin

- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

GetPersonalizedRanking

Service : Amazon Personalize Runtime

Re-ranks une liste d'éléments recommandés pour l'utilisateur donné. Le premier élément de la liste est considéré comme l'élément le plus susceptible d'intéresser l'utilisateur.

> **Note**
>
> La solution utilisée pour la campagne doit avoir été créée à l'aide d'une recette de type PERSONALIZED_RANKING.

Syntaxe de la demande

```
POST /personalize-ranking HTTP/1.1
Content-type: application/json

{
   "campaignArn": "string",
   "context": {
      "string" : "string"
   },
   "filterArn": "string",
   "filterValues": {
      "string" : "string"
   },
   "inputList": [ "string" ],
   "metadataColumns": {
      "string" : [ "string" ]
   },
   "userId": "string"
}
```

Paramètres de demande URI

La demande n’utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Le nom de ressource Amazon (ARN) de la campagne à utiliser pour générer le classement personnalisé.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : oui

context

Les métadonnées contextuelles à utiliser pour obtenir des recommandations. Les métadonnées contextuelles incluent toutes les informations d'interaction susceptibles d'être pertinentes pour obtenir les recommandations d'un utilisateur, telles que la position actuelle de l'utilisateur ou le type d'appareil.

Type : mappage chaîne/chaîne

Entrées sur la carte : nombre maximum de 150 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 150.

Modèle de clé :`[A-Za-z\d_]+`

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

filterArn

Le nom de ressource Amazon (ARN) d'un filtre que vous avez créé pour inclure des éléments ou exclure des éléments des recommandations destinées à un utilisateur donné. Pour de plus amples informations, veuillez consulter Filtrage des recommandations.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterValues

Les valeurs à utiliser lors du filtrage des recommandations. Pour chaque paramètre d'espace réservé dans votre expression de filtre, indiquez le nom du paramètre (dans les cas correspondants) en tant que clé et la ou les valeurs du filtre en tant que valeur correspondante. Séparez les valeurs multiples d'un paramètre par une virgule.

Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des éléments, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des articles, vous pouvez omettre le `filter-values` .Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

Pour de plus amples informations, veuillez consulter Filtrage des recommandations.

Type : mappage chaîne/chaîne

Entrées sur la carte : nombre maximum de 25 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 50.

Modèle de clé :`[A-Za-z0-9_]+`

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

inputList

Une liste d'éléments (`paritemId`) à classer. Si un élément n'était pas inclus dans l'ensemble de données d'entraînement, il est ajouté à la fin de la liste reclassée. Si vous incluez des métadonnées dans les recommandations, le maximum est de 50. Sinon, le maximum est de 500.

Type : tableau de chaînes

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

metadataColumns

Si vous avez activé les métadonnées dans les recommandations lorsque vous avez créé ou mis à jour la campagne, spécifiez les colonnes de métadonnées de votre jeu de données Items à inclure dans le classement personnalisé. La clé de la carte est, `ITEMS` et sa valeur est, une liste de noms

de colonnes de votre jeu de données Items. Le nombre maximum de colonnes que vous pouvez fournir est de 10.

Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultez la section Activation des métadonnées dans les recommandations relatives à une campagne.

TYpe : mappage de chaîne à tableau de chaînes

Entrées cartographiques : nombre maximum d'un élément.

Contraintes relatives à la longueur des touches : longueur maximale de 256.

Membres du tableau : nombre maximum de 99 éléments.

Contraintes de longueur : longueur maximale de 150.

Obligatoire : non

userId

L'utilisateur pour lequel vous souhaitez que la campagne fournisse un classement personnalisé.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "personalizedRanking": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
```

```
      }
   ],
   "recommendationId": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

personalizedRanking

Liste des éléments les plus susceptibles d'intéresser l'utilisateur. Le maximum est de 500.

Type : tableau d'objets PredictedItem

recommendationId

L'ID de la recommandation.

Type : Chaîne

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

La ressource spécifiée n'existe pas.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2

- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

GetRecommendations

Service : Amazon Personalize Runtime

Renvoie une liste d'articles recommandés. Pour les campagnes, le nom de ressource Amazon (ARN) de la campagne est obligatoire et la saisie des utilisateurs et des articles requise dépend du type de recette utilisé pour créer la solution qui soutient la campagne, comme suit :

- USER_PERSONALIZATION - `userId` obligatoire, non utilisé `itemId`
- RELATED_ITEMS - `itemId` obligatoire, non utilisé `userId`

> **Note**
>
> Les campagnes soutenues par une solution créée à l'aide d'une recette de type PERSONALIZED_RANKING utilisent l'API. GetPersonalizedRanking

Pour les recommandateurs, l'ARN du recommandateur est obligatoire et l'élément requis et les données saisies par l'utilisateur dépendent du cas d'utilisation (recette basée sur le domaine) qui sous-tend le recommandateur. Pour plus d'informations sur les exigences des cas d'utilisation, consultez la section Choix des cas d'utilisation recommandés.

Syntaxe de la demande

```
POST /recommendations HTTP/1.1
Content-type: application/json

{
   "campaignArn": "string",
   "context": {
      "string" : "string"
   },
   "filterArn": "string",
   "filterValues": {
      "string" : "string"
   },
   "itemId": "string",
   "metadataColumns": {
      "string" : [ "string" ]
   },
   "numResults": number,
```

```
   "promotions": [
      {
         "filterArn": "string",
         "filterValues": {
            "string" : "string"
         },
         "name": "string",
         "percentPromotedItems": number
      }
   ],
   "recommenderArn": "string",
   "userId": "string"
}
```

Paramètres de demande URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

campaignArn

Le nom de ressource Amazon (ARN) de la campagne à utiliser pour obtenir des recommandations.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

context

Les métadonnées contextuelles à utiliser pour obtenir des recommandations. Les métadonnées contextuelles incluent toutes les informations d'interaction susceptibles d'être pertinentes pour obtenir les recommandations d'un utilisateur, telles que la position actuelle de l'utilisateur ou le type d'appareil.

Type : mappage chaîne/chaîne

Entrées sur la carte : nombre maximum de 150 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 150.

Modèle de clé :`[A-Za-z\d_]+`

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

filterArn

L'ARN du filtre à appliquer aux recommandations renvoyées. Pour de plus amples informations, veuillez consulter Filtrage des recommandations.

Lorsque vous utilisez ce paramètre, assurez-vous que la ressource de filtre l'est`ACTIVE`.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterValues

Les valeurs à utiliser lors du filtrage des recommandations. Pour chaque paramètre d'espace réservé dans votre expression de filtre, indiquez le nom du paramètre (dans les cas correspondants) en tant que clé et la ou les valeurs du filtre en tant que valeur correspondante. Séparez les valeurs multiples d'un paramètre par une virgule.

Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des éléments, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des articles, vous pouvez omettre le `filter-values` .Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

Pour plus d'informations, consultez la section Filtrage des recommandations et des segments d'utilisateurs.

Type : mappage chaîne/chaîne

Entrées sur la carte : nombre maximum de 25 éléments.

Contraintes relatives à la longueur des touches : longueur maximale de 50.

Modèle de clé :`[A-Za-z0-9_]+`

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

itemId

L'identifiant de l'article pour lequel fournir des recommandations.

Obligatoire pour le type de `RELATED_ITEMS` recette.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

metadataColumns

Si vous avez activé les métadonnées dans les recommandations lors de la création ou de la mise à jour de la campagne ou de la recommandation, spécifiez les colonnes de métadonnées de votre jeu de données Items à inclure dans les recommandations d'articles. La clé de la carte est, `ITEMS` et sa valeur est, une liste de noms de colonnes de votre jeu de données Items. Le nombre maximum de colonnes que vous pouvez fournir est de 10.

Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultez la section Activation des métadonnées dans les recommandations relatives à une campagne. Pour plus d'informations sur l'activation des métadonnées pour un recommandataire, voir Activation des métadonnées dans les recommandations destinées à un recommandateur.

TYpe : mappage de chaîne à tableau de chaînes

Entrées cartographiques : nombre maximum d'un élément.

Contraintes relatives à la longueur des touches : longueur maximale de 256.

Membres du tableau : nombre maximum de 99 éléments.

Contraintes de longueur : longueur maximale de 150.

Obligatoire : non

numResults

Le nombre de résultats à renvoyer. La valeur par défaut est 25. Si vous incluez des métadonnées dans les recommandations, le maximum est de 50. Sinon, le maximum est de 500.

Type : Integer

Plage valide : Valeur minimum de 0.

Obligatoire : non

promotions

Les promotions à appliquer à la demande de recommandation. Une promotion définit des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés.

Type : tableau d'objets Promotion

Membres du tableau : nombre maximum de 1 élément.

Obligatoire : non

recommenderArn

Le nom de ressource Amazon (ARN) du recommandateur à utiliser pour obtenir des recommandations. Fournissez un ARN de recommandation si vous avez créé un groupe de jeux de données de domaine avec un outil de recommandation pour un cas d'utilisation de domaine.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

userId

L'ID utilisateur pour lequel fournir des recommandations.

Obligatoire pour le type de `USER_PERSONALIZATION` recette.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "itemList": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

itemList

Liste de recommandations classées par ordre décroissant par score de prédiction. Il peut y avoir un maximum de 500 éléments dans la liste.

Type : tableau d'objets PredictedItem

recommendationId

L'ID de la recommandation.

Type : Chaîne

Erreurs

InvalidInputException

Fournissez une valeur valide pour le champ ou le paramètre.

Code d'état HTTP : 400

ResourceNotFoundException

La ressource spécifiée n'existe pas.

Code d'état HTTP :404

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS kits SDK spécifiques à une langue, consultez les rubriques suivantes :

- AWS Interface de ligne de commande V2
- AWS SDK pour .NET V4
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour PHP V3
- AWS SDK pour Python (Boto3)
- AWS SDK pour Ruby V3

Types de données

Les types de données suivants sont pris en charge par Amazon Personalize :

- Algorithm
- AlgorithmImage
- AutoMLConfig

- AutoMLResult
- AutoTrainingConfig
- BatchInferenceJob
- BatchInferenceJobConfig
- BatchInferenceJobInput
- BatchInferenceJobOutput
- BatchInferenceJobSummary
- BatchSegmentJob
- BatchSegmentJobInput
- BatchSegmentJobOutput
- BatchSegmentJobSummary
- Campaign
- CampaignConfig
- CampaignSummary
- CampaignUpdateSummary
- CategoricalHyperParameterRange
- ContinuousHyperParameterRange
- DataDeletionJob
- DataDeletionJobSummary
- Dataset
- DatasetExportJob
- DatasetExportJobOutput
- DatasetExportJobSummary
- DatasetGroup
- DatasetGroupSummary
- DatasetImportJob
- DatasetImportJobSummary
- DatasetSchema
- DatasetSchemaSummary
- DatasetSummary

- DatasetUpdateSummary
- DataSource
- DefaultCategoricalHyperParameterRange
- DefaultContinuousHyperParameterRange
- DefaultHyperParameterRanges
- DefaultIntegerHyperParameterRange
- EventParameters
- EventsConfig
- EventTracker
- EventTrackerSummary
- FeatureTransformation
- FieldsForThemeGeneration
- Filter
- FilterSummary
- HPOConfig
- HPOObjective
- HPOResourceConfig
- HyperParameterRanges
- IntegerHyperParameterRange
- MetricAttribute
- MetricAttribution
- MetricAttributionOutput
- MetricAttributionSummary
- OptimizationObjective
- Recipe
- RecipeSummary
- Recommender
- RecommenderConfig
- RecommenderSummary
- RecommenderUpdateSummary

- S3DataConfig
- Solution
- SolutionConfig
- SolutionSummary
- SolutionUpdateConfig
- SolutionUpdateSummary
- SolutionVersion
- SolutionVersionSummary
- Tag
- ThemeGenerationConfig
- TrainingDataConfig
- TunedHPOParams

Les types de données suivants sont pris en charge par Amazon Personalize Events :

- Action
- ActionInteraction
- Event
- Item
- MetricAttribution
- User

Les types de données suivants sont pris en charge par Amazon Personalize Runtime :

- PredictedAction
- PredictedItem
- Promotion

Amazon Personalize

Les types de données suivants sont pris en charge par Amazon Personalize :

- Algorithm

- AlgorithmImage
- AutoMLConfig
- AutoMLResult
- AutoTrainingConfig
- BatchInferenceJob
- BatchInferenceJobConfig
- BatchInferenceJobInput
- BatchInferenceJobOutput
- BatchInferenceJobSummary
- BatchSegmentJob
- BatchSegmentJobInput
- BatchSegmentJobOutput
- BatchSegmentJobSummary
- Campaign
- CampaignConfig
- CampaignSummary
- CampaignUpdateSummary
- CategoricalHyperParameterRange
- ContinuousHyperParameterRange
- DataDeletionJob
- DataDeletionJobSummary
- Dataset
- DatasetExportJob
- DatasetExportJobOutput
- DatasetExportJobSummary
- DatasetGroup
- DatasetGroupSummary
- DatasetImportJob
- DatasetImportJobSummary
- DatasetSchema

- DatasetSchemaSummary
- DatasetSummary
- DatasetUpdateSummary
- DataSource
- DefaultCategoricalHyperParameterRange
- DefaultContinuousHyperParameterRange
- DefaultHyperParameterRanges
- DefaultIntegerHyperParameterRange
- EventParameters
- EventsConfig
- EventTracker
- EventTrackerSummary
- FeatureTransformation
- FieldsForThemeGeneration
- Filter
- FilterSummary
- HPOConfig
- HPOObjective
- HPOResourceConfig
- HyperParameterRanges
- IntegerHyperParameterRange
- MetricAttribute
- MetricAttribution
- MetricAttributionOutput
- MetricAttributionSummary
- OptimizationObjective
- Recipe
- RecipeSummary
- Recommender
- RecommenderConfig

- RecommenderSummary
- RecommenderUpdateSummary
- S3DataConfig
- Solution
- SolutionConfig
- SolutionSummary
- SolutionUpdateConfig
- SolutionUpdateSummary
- SolutionVersion
- SolutionVersionSummary
- Tag
- ThemeGenerationConfig
- TrainingDataConfig
- TunedHPOParams

Algorithm

Service : Amazon Personalize

Décrit un algorithme personnalisé.

Table des matières

algorithmArn

Le nom de ressource Amazon (ARN) de l'algorithme.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

algorithmImage

L'URI du conteneur Docker pour l'image de l'algorithme.

Type : objet AlgorithmImage

Obligatoire : non

creationDateTime

Date et heure (en heure Unix) auxquelles l'algorithme a été créé.

Type : Timestamp

Obligatoire : non

defaultHyperParameterRanges

Spécifie les hyperparamètres par défaut, leurs plages et indique s'ils sont réglables. La valeur d'un hyperparamètre réglable peut être déterminée lors de l'optimisation des hyperparamètres (HPO).

Type : objet DefaultHyperParameterRanges

Obligatoire : non

defaultHyperParameters

Spécifie les hyperparamètres par défaut.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

defaultResourceConfig

Spécifie le nombre maximal par défaut de tâches de formation et de tâches d'entraînement parallèles.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles l'algorithme a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom de l'algorithme.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

roleArn

L'Amazon Resource Name (ARN) du rôle.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

trainingInputMode

Le mode de saisie d'entraînement.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

AlgorithmImage

Service : Amazon Personalize

Décrit une image d'algorithme.

Table des matières

dockerURI

L'URI du conteneur Docker pour l'image de l'algorithme.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

name

Nom de l'image de l'algorithme.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

AutoMLConfig

Service : Amazon Personalize

Lorsque la solution exécute AutoML (`performAutoMLc`'est vrai dans CreateSolution), Amazon Personalize détermine quelle recette, à partir de la liste spécifiée, optimise la métrique donnée. Amazon Personalize utilise ensuite cette recette pour la solution.

Table des matières

metricName

Métrique à optimiser.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

recipeList

Liste des recettes candidates.

Type : tableau de chaînes

Membres du tableau : nombre maximum de 100 éléments.

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

AutoMLResult

Service : Amazon Personalize

Lorsque la solution exécute AutoML (`performAutoMLest` vrai dans CreateSolution), spécifie la recette qui a le mieux optimisé la métrique spécifiée.

Table des matières

bestRecipeArn

Le nom de ressource Amazon (ARN) de la meilleure recette.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

AutoTrainingConfig

Service : Amazon Personalize

La configuration d'entraînement automatique à utiliser lorsque `performAutoTraining` c'est vrai.

Table des matières

schedulingExpression

Spécifie la fréquence à laquelle les nouvelles versions de solution doivent être automatiquement entraînées. Spécifiez une expression de taux au format taux (unité de valeur). Pour la valeur, spécifiez un nombre compris entre 1 et 30. Pour unité, spécifiez `day` ou`days`. Par exemple, pour créer automatiquement une nouvelle version de solution tous les 5 jours, spécifiez`rate(5 days)`. La valeur par défaut est tous les 7 jours.

Pour plus d'informations sur l'entraînement automatique, consultez la section Création et configuration d'une solution.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 16

Modèle : `rate\(\d+ days?\)`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchInferenceJob

Service : Amazon Personalize

Contient des informations sur une tâche d'inférence par lots.

Table des matières

batchInferenceJobArn

L'Amazon Resource Name (ARN) de la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

batchInferenceJobConfig

Un mappage de chaîne à chaîne des détails de configuration d'une tâche d'inférence par lots.

Type : objet BatchInferenceJobConfig

Obligatoire : non

batchInferenceJobMode

Le mode du job.

Type : String

Valeurs valides : `BATCH_INFERENCE | THEME_GENERATION`

Obligatoire : non

creationDateTime

Heure à laquelle la tâche d'inférence par lots a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si la tâche d'inférence par lots a échoué, raison de l'échec.

Type : chaîne

Obligatoire : non

filterArn

L'ARN du filtre utilisé lors de la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

jobInput

Le chemin Amazon S3 qui mène aux données d'entrée utilisées pour générer la tâche d'inférence par lots.

Type : objet BatchInferenceJobInput

Obligatoire : non

jobName

Nom de la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

jobOutput

Le compartiment Amazon S3 qui contient les données de sortie générées par la tâche d'inférence par lots.

Type : objet BatchInferenceJobOutput

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle la tâche d'inférence par lots a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

numResults

Nombre de recommandations générées par la tâche d'inférence par lots. Ce numéro inclut les messages d'erreur générés en cas d'échec des enregistrements de saisie.

Type : entier

Obligatoire : non

roleArn

L'ARN du rôle Amazon Identity and Access Management (IAM) qui a demandé la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

solutionVersionArn

Nom de ressource Amazon (ARN) de la version de solution à partir de laquelle la tâche d'inférence par lots a été créée.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche d'inférence par lots. Le statut est l'une des valeurs suivantes :

- PENDING
- EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

themeGenerationConfig

Les paramètres de génération du thème de la tâche.

Type : objet ThemeGenerationConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchInferenceJobConfig

Service : Amazon Personalize

Détails de configuration d'une tâche d'inférence par lots.

Table des matières

itemExplorationConfig

Une carte de chaîne à chaîne spécifiant les hyperparamètres de configuration d'exploration, y compris `explorationWeight` et `explorationItemAgeCutOff` que vous souhaitez utiliser pour configurer le volume d'exploration d'articles utilisé par Amazon Personalize lorsqu'il recommande des articles. Voir Personnalisation utilisateur.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

rankingInfluence

Une carte du classement influence les valeurs de POPULARITÉ et de FRAÎCHEUR. Pour chaque clé, spécifiez une valeur numérique comprise entre 0,0 et 1,0 qui détermine l'influence de ce facteur de classement sur les recommandations finales. Une valeur plus proche de 1,0 donne plus de poids au facteur, tandis qu'une valeur plus proche de 0,0 réduit son influence. Si ce n'est pas spécifié, la valeur par défaut des deux est 0,0.

Type : mappage chaîne à double

Clés valides : `POPULARITY | FRESHNESS`

Plage valide : Valeur minimum de 0. Valeur maximale de 1.

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

BatchInferenceJobInput

Service : Amazon Personalize

Configuration d'entrée d'une tâche d'inférence par lots.

Table des matières

s3DataSource

L'URI de l'emplacement Amazon S3 qui contient vos données d'entrée. Le compartiment Amazon S3 doit se trouver dans la même région que le point de terminaison de l'API que vous appelez.

Type : objet S3DataConfig

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchInferenceJobOutput

Service : Amazon Personalize

Les paramètres de configuration de sortie d'une tâche d'inférence par lots.

Table des matières

s3DataDestination

Informations sur le compartiment Amazon S3 dans lequel est stockée la sortie de la tâche d'inférence par lots.

Type : objet S3DataConfig

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchInferenceJobSummary

Service : Amazon Personalize

Une version tronquée du. BatchInferenceJob L'ListBatchInferenceJobsopération renvoie une liste de résumés des tâches d'inférence par lots.

Table des matières

batchInferenceJobArn

L'Amazon Resource Name (ARN) de la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

batchInferenceJobMode

Le mode du job.

Type : String

Valeurs valides : `BATCH_INFERENCE | THEME_GENERATION`

Obligatoire : non

creationDateTime

Heure à laquelle la tâche d'inférence par lots a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si la tâche d'inférence par lots a échoué, raison de l'échec.

Type : chaîne

Obligatoire : non

jobName

Nom de la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle la tâche d'inférence par lots a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

solutionVersionArn

L'ARN de la version de solution utilisée par la tâche d'inférence par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche d'inférence par lots. Le statut est l'une des valeurs suivantes :

- PENDING
- EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchSegmentJob

Service : Amazon Personalize

Contient des informations sur une tâche de segmentation par lots.

Table des matières

batchSegmentJobArn

Le nom de ressource Amazon (ARN) de la tâche de segment par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

creationDateTime

Heure à laquelle la tâche de segment par lots a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si le travail de segmentation par lots a échoué, raison de l'échec.

Type : chaîne

Obligatoire : non

filterArn

L'ARN du filtre utilisé lors de la tâche de segmentation par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

jobInput

Le chemin Amazon S3 qui mène aux données d'entrée utilisées pour générer la tâche de segmentation par lots.

Type : objet BatchSegmentJobInput

Obligatoire : non

jobName

Nom de la tâche de segmentation par lots.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

jobOutput

Le compartiment Amazon S3 qui contient les données de sortie générées par la tâche de segmentation par lots.

Type : objet BatchSegmentJobOutput

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle la tâche de segmentation par lots a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

numResults

Nombre d'utilisateurs prédits générés par la tâche de segmentation par lots pour chaque ligne de données d'entrée. Le nombre maximum d'utilisateurs par segment est de 5 millions.

Type : entier

Obligatoire : non

roleArn

L'ARN du rôle Amazon Identity and Access Management (IAM) qui a demandé la tâche de segmentation par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

solutionVersionArn

Nom de ressource Amazon (ARN) de la version de solution utilisée par la tâche de segmentation par lots pour générer des segments par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche de segmentation par lots. Le statut est l'une des valeurs suivantes :

- PENDING
- EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchSegmentJobInput

Service : Amazon Personalize

Configuration d'entrée d'une tâche de segmentation par lots.

Table des matières

s3DataSource

Les détails de configuration d'un compartiment d'entrée ou de sortie Amazon S3.

Type : objet S3DataConfig

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchSegmentJobOutput

Service : Amazon Personalize

Les paramètres de configuration de sortie d'une tâche de segment par lots.

Table des matières

s3DataDestination

Les détails de configuration d'un compartiment d'entrée ou de sortie Amazon S3.

Type : objet S3DataConfig

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

BatchSegmentJobSummary

Service : Amazon Personalize

Version tronquée du type de BatchSegmentJobdonnées. ListBatchSegmentJobsL'opération renvoie une liste de résumés de tâches par segments par lots.

Table des matières

batchSegmentJobArn

Le nom de ressource Amazon (ARN) de la tâche de segment par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

creationDateTime

Heure à laquelle la tâche de segment par lots a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si le travail de segmentation par lots a échoué, raison de l'échec.

Type : chaîne

Obligatoire : non

jobName

Nom de la tâche de segmentation par lots.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle la tâche de segmentation par lots a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

solutionVersionArn

Nom de ressource Amazon (ARN) de la version de solution utilisée par la tâche de segmentation par lots pour générer des segments par lots.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche de segmentation par lots. Le statut est l'une des valeurs suivantes :

- PENDING
- EN COURS
- ACTIF
- ÉCHEC DE LA CRÉATION

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Campaign

Service : Amazon Personalize

Objet qui décrit le déploiement d'une version de solution. Pour plus d'informations sur les campagnes, consultez CreateCampaign.

Table des matières

campaignArn

Amazon Resource Name (ARN) de la campagne.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

campaignConfig

Les détails de configuration d'une campagne.

Type : objet CampaignConfig

Obligatoire : non

creationDateTime

Date et heure (au format Unix) auxquelles la campagne a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si une campagne échoue, la raison de cet échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (au format Unix) auxquelles la campagne a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

latestCampaignUpdate

Fournit un résumé des propriétés d'une mise à jour de campagne. Pour obtenir une liste complète, appelez l'DescribeCampaignAPI.

> **Note**
>
> Le `latestCampaignUpdate` champ n'est renvoyé que lorsque la campagne a reçu au moins un `UpdateCampaign` appel.

Type : objet CampaignUpdateSummary

Obligatoire : non

minProvisionedTPS

Spécifie le nombre minimum de transactions provisionnées demandées (recommandations) par seconde. Un montant élevé `minProvisionedTPS` fera augmenter votre facture. Nous vous recommandons de commencer par 1 pour `minProvisionedTPS` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et augmentez-les `minProvisionedTPS` si nécessaire.

Type : entier

Plage valide : Valeur minimum de 1.

Obligatoire : non

name

Nom de la campagne.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de solution utilisée par la campagne.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la campagne.

Une campagne peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

CampaignConfig

Service : Amazon Personalize

Les détails de configuration d'une campagne.

Table des matières

enableMetadataWithRecommendations

Si les métadonnées contenant des recommandations sont activées pour la campagne. Si cette option est activée, vous pouvez spécifier les colonnes de votre jeu de données Items dans votre demande de recommandations. Amazon Personalize renvoie ces données pour chaque article dans la réponse de recommandation. Pour plus d'informations sur l'activation des métadonnées pour une campagne, consultez la section Activation des métadonnées dans les recommandations relatives à une campagne.

Si vous activez les métadonnées dans les recommandations, vous devrez payer des frais supplémentaires. Pour plus d'informations, consultez les tarifs d'Amazon Personalize.

Type : booléen

Obligatoire : non

itemExplorationConfig

Spécifie les hyperparamètres de configuration d'exploration, notamment `explorationWeight` et `explorationItemAgeCutOff` que vous souhaitez utiliser pour configurer la quantité d'objets explorés par Amazon Personalize lorsqu'il recommande des articles. Fournissez `itemExplorationConfig` des données uniquement si votre solution utilise la recette de personnalisation utilisateur.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

rankingInfluence

Une carte du classement influence les valeurs de POPULARITÉ et de FRAÎCHEUR. Pour chaque clé, spécifiez une valeur numérique comprise entre 0,0 et 1,0 qui détermine l'influence de ce facteur de classement sur les recommandations finales. Une valeur plus proche de 1,0 donne plus de poids au facteur, tandis qu'une valeur plus proche de 0,0 réduit son influence. Si ce n'est pas spécifié, la valeur par défaut des deux est 0,0.

Type : mappage chaîne à double

Clés valides : `POPULARITY | FRESHNESS`

Plage valide : Valeur minimum de 0. Valeur maximale de 1.

Obligatoire : non

syncWithLatestSolutionVersion

Si la campagne est automatiquement mise à jour pour utiliser la dernière version de solution (modèle entraîné) d'une solution. Si vous le spécifiez`True`, vous devez spécifier l'ARN de votre solution pour le `SolutionVersionArn` paramètre. Il doit être au `SolutionArn/$LATEST` format. La valeur par défaut est `False` et vous devez mettre à jour manuellement la campagne pour déployer la dernière version de la solution.

Pour plus d'informations sur les mises à jour automatiques des campagnes, voir Activation des mises à jour automatiques des campagnes.

Type : booléen

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

CampaignSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une campagne. Pour obtenir une liste complète, appelez l'DescribeCampaignAPI.

Table des matières

campaignArn

Amazon Resource Name (ARN) de la campagne.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

creationDateTime

Date et heure (en heure Unix) auxquelles la campagne a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si une campagne échoue, la raison de cet échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la campagne a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom de la campagne.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État de la campagne.

Une campagne peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

CampaignUpdateSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une mise à jour de campagne. Pour obtenir une liste complète, appelez l'DescribeCampaignAPI.

Table des matières

campaignConfig

Les détails de configuration d'une campagne.

Type : objet CampaignConfig

Obligatoire : non

creationDateTime

Date et heure (en heure Unix) auxquelles la mise à jour de la campagne a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si la mise à jour d'une campagne échoue, la raison de cet échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la dernière mise à jour de la campagne a été mise à jour.

Type : Timestamp

Obligatoire : non

minProvisionedTPS

Spécifie le nombre minimum de transactions provisionnées demandées (recommandations) par seconde qu'Amazon Personalize prendra en charge.

Type : entier

Plage valide : Valeur minimum de 1.

Obligatoire : non

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de solution déployée.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la mise à jour de la campagne.

Une mise à jour de campagne peut se produire dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

CategoricalHyperParameterRange

Service : Amazon Personalize

Fournit le nom et la plage d'un hyperparamètre de type catégorie.

Table des matières

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

values

Liste des catégories de l'hyperparamètre.

Type : tableau de chaînes

Membres du tableau : nombre maximum de 100 éléments.

Contraintes de longueur : Longueur maximum de 1 000.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ContinuousHyperParameterRange

Service : Amazon Personalize

Fournit le nom et la plage d'un hyperparamètre de type continu.

Table des matières

maxValue

Valeur maximale autorisée pour l'hyperparamètre.

Type : double

Plage valide : valeur minimale de -1000000.

Obligatoire : non

minValue

Valeur minimale autorisée pour l'hyperparamètre.

Type : double

Plage valide : valeur minimale de -1000000.

Obligatoire : non

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

DataDeletionJob

Service : Amazon Personalize

Décrit une tâche qui supprime par lots toutes les références à des utilisateurs spécifiques d'un groupe de jeux de données Amazon Personalize. Pour plus d'informations sur la création d'une tâche de suppression de données, consultez la section Suppression d'utilisateurs.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la tâche de suppression des données.

Type : Timestamp

Obligatoire : non

dataDeletionJobArn

Le nom de ressource Amazon (ARN) de la tâche de suppression des données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de données dont la tâche supprime les enregistrements.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

dataSource

Décrit la source de données qui contient les données à charger dans un ensemble de données ou la liste des enregistrements à supprimer d'Amazon Personalize.

Type : objet DataSource

Obligatoire : non

failureReason

Si une tâche de suppression de données échoue, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

jobName

Nom de la tâche de suppression des données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) de la dernière mise à jour de la tâche de suppression des données.

Type : Timestamp

Obligatoire : non

numDeleted

Le nombre d'enregistrements supprimés par une tâche TERMINÉE.

Type : entier

Obligatoire : non

roleArn

Le nom de ressource Amazon (ARN) du rôle IAM autorisé à lire depuis la source de données Amazon S3.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

status

État de la tâche de suppression des données.

Une tâche de suppression de données peut avoir l'un des statuts suivants :

- EN ATTENTE > EN COURS > TERMINÉ OU ÉCHEC

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DataDeletionJobSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une tâche de suppression de données. Pour obtenir une liste complète, appelez l'opération DescribeDataDeletionJobAPI.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la tâche de suppression des données.

Type : Timestamp

Obligatoire : non

dataDeletionJobArn

Le nom de ressource Amazon (ARN) de la tâche de suppression des données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetGroupArn

Nom de ressource Amazon (ARN) du groupe de données dont la tâche a supprimé des enregistrements.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si une tâche de suppression de données échoue, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

jobName

Nom de la tâche de suppression des données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) de la dernière mise à jour de la tâche de suppression des données.

Type : Timestamp

Obligatoire : non

status

État de la tâche de suppression des données.

Une tâche de suppression de données peut avoir l'un des statuts suivants :

- EN ATTENTE > EN COURS > TERMINÉ OU ÉCHEC

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

Dataset

Service : Amazon Personalize

Fournit des métadonnées pour un ensemble de données.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) du jeu de données.

Type : Timestamp

Obligatoire : non

datasetArn

Le nom de ressource Amazon (ARN) de l'ensemble de données pour lequel vous souhaitez obtenir des métadonnées.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetType

L’une des valeurs suivantes :

- Interactions

- Éléments
- Utilisateurs
- Actions
- Action_Interactions

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `^[A-Za-z_]+$`

Obligatoire : non

lastUpdatedDateTime

Horodatage indiquant la date de mise à jour du jeu de données.

Type : Timestamp

Obligatoire : non

latestDatasetUpdate

Décrit la dernière mise à jour de l'ensemble de données.

Type : objet DatasetUpdateSummary

Obligatoire : non

name

Nom de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

schemaArn

ARN du schéma associé.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de l'ensemble de données.

Un jeu de données peut avoir l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

trackingId

L'ID du suivi des événements pour un ensemble de données d'interactions Action. Vous spécifiez l'ID du tracker dans le fonctionnement de `PutActionInteractions` l'API. Amazon Personalize l'utilise pour diriger les nouvelles données vers le jeu de données Action interactions de votre groupe de jeux de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2

- AWSSDK pour Ruby V3

DatasetExportJob

Service : Amazon Personalize

Décrit une tâche qui exporte un ensemble de données vers un compartiment Amazon S3. Pour de plus amples informations, veuillez consulter CreateDatasetExportJob.

Une tâche d'exportation d'un ensemble de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la tâche d'exportation de l'ensemble de données.

Type : Timestamp

Obligatoire : non

datasetArn

Le nom de ressource Amazon (ARN) de l'ensemble de données à exporter.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetExportJobArn

Le nom de ressource Amazon (ARN) de la tâche d'exportation de l'ensemble de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

En cas d'échec d'une tâche d'exportation d'un ensemble de données, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

ingestionMode

Les données à exporter, en fonction de la façon dont vous les avez importées. Vous pouvez choisir d'exporter les `BULK` données que vous avez importées à l'aide d'une tâche d'importation de jeux de `PUT` données, les données que vous avez importées de manière incrémentielle (à l'aide de la console PutUsers et PutItems des opérations), ou `ALL` pour les deux types. PutEvents La valeur par défaut est `PUT`.

Type : String

Valeurs valides : `BULK | PUT | ALL`

Obligatoire : non

jobName

Nom de la tâche d'exportation.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

jobOutput

Le chemin d'accès au compartiment Amazon S3 dans lequel les résultats de la tâche sont stockés. Par exemple :

```
s3://bucket-name/folder-name/
```

Type : objet DatasetExportJobOutput

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le statut de la tâche d'exportation de l'ensemble de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

roleArn

Le nom de ressource Amazon (ARN) du rôle de service IAM autorisé à ajouter des données à votre compartiment Amazon S3 de sortie.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche d'exportation de l'ensemble de données.

Une tâche d'exportation d'un ensemble de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetExportJobOutput

Service : Amazon Personalize

Les paramètres de configuration de sortie d'une tâche d'exportation d'un ensemble de données.

Table des matières

s3DataDestination

Les détails de configuration d'un compartiment d'entrée ou de sortie Amazon S3.

Type : objet S3DataConfig

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetExportJobSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une tâche d'exportation d'un ensemble de données. Pour obtenir une liste complète, appelez l'DescribeDatasetExportJobAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles la tâche d'exportation de l'ensemble de données a été créée.

Type : Timestamp

Obligatoire : non

datasetExportJobArn

Nom de ressource Amazon (ARN) de la tâche d'exportation de l'ensemble de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si une tâche d'exportation d'un ensemble de données échoue, raison de cet échec.

Type : chaîne

Obligatoire : non

jobName

Nom de la tâche d'exportation de l'ensemble de données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le statut de la tâche d'exportation de l'ensemble de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

status

État de la tâche d'exportation de l'ensemble de données.

Une tâche d'exportation d'un ensemble de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetGroup

Service : Amazon Personalize

Un groupe de jeux de données est un ensemble d'ensembles de données connexes (interactions entre éléments, utilisateurs, éléments, actions, interactions entre actions). Vous créez un groupe de jeux de données en appelant CreateDatasetGroup. Vous créez ensuite un ensemble de données et vous l'ajoutez à un groupe de jeux de données en appelant CreateDataset. Le groupe de jeux de données est utilisé pour créer et entraîner une solution en appelant CreateSolution. Un groupe d'ensembles de données ne peut contenir qu'un seul ensemble de données de chaque type.

Vous pouvez spécifier une clé AWS Key Management Service (KMS) pour chiffrer les ensembles de données du groupe.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) du groupe de jeux de données.

Type : Timestamp

Obligatoire : non

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

domain

Domaine d'un groupe de jeux de données de domaine.

Type : String

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

failureReason

Si la création d'un groupe de jeux de données échoue, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

kmsKeyArn

Le nom de ressource Amazon (ARN) de la clé AWS Key Management Service (KMS) utilisée pour chiffrer les ensembles de données.

Type : String

Contraintes de longueur : longueur maximale de 2048.

Modèle : `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatoire : non

lastUpdatedDateTime

Date et heure de la dernière mise à jour (en heure Unix) du groupe de jeux de données.

Type : Timestamp

Obligatoire : non

name

Le nom du groupe de jeux de données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

roleArn

L'ARN du rôle Gestion des identités et des accès AWS (IAM) autorisé à accéder à la clé AWS Key Management Service (KMS). La fourniture d'un rôle IAM n'est valide que si vous spécifiez également une clé KMS.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : non

status

État actuel du groupe de jeux de données.

Un groupe de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- SUPPRIMER EN ATTENTE

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetGroupSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'un groupe de jeux de données. Pour obtenir une liste complète, appelez l'DescribeDatasetGroupAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles le groupe de jeux de données a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

domain

Domaine d'un groupe de jeux de données de domaine.

Type : String

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

failureReason

Si la création d'un groupe de jeux de données échoue, la raison de cet échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le groupe de jeux de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom du groupe de jeux de données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État du groupe de jeux de données.

Un groupe de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- SUPPRIMER EN ATTENTE

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

DatasetImportJob

Service : Amazon Personalize

Décrit une tâche qui importe des données d'entraînement depuis une source de données (compartiment Amazon S3) vers un jeu de données Amazon Personalize. Pour de plus amples informations, veuillez consulter CreateDatasetImportJob.

Une tâche d'importation de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la tâche d'importation de l'ensemble de données.

Type : Timestamp

Obligatoire : non

datasetArn

L'Amazon Resource Name (ARN) du jeu de données qui reçoit les données importées.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetImportJobArn

L'ARN de la tâche d'importation du jeu de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

dataSource

Le compartiment Amazon S3 qui contient les données d'entraînement à importer.

Type : objet DataSource

Obligatoire : non

failureReason

En cas d'échec d'une tâche d'importation d'un ensemble de données, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

importMode

Mode d'importation utilisé par la tâche d'importation de l'ensemble de données pour importer de nouveaux enregistrements.

Type : String

Valeurs valides : `FULL | INCREMENTAL`

Obligatoire : non

jobName

Le nom de la tâche d'importation.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le jeu de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

publishAttributionMetricsToS3

Si la tâche publie des métriques sur Amazon S3 pour une attribution de métriques.

Type : booléen

Obligatoire : non

roleArn

L'ARN du rôle IAM autorisé à lire à partir de la source de données Amazon S3.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la tâche d'importation de l'ensemble de données.

Une tâche d'importation de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetImportJobSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une tâche d'importation de jeux de données. Pour obtenir une liste complète, appelez l'DescribeDatasetImportJobAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles la tâche d'importation de l'ensemble de données a été créée.

Type : Timestamp

Obligatoire : non

datasetImportJobArn

Le nom de ressource Amazon (ARN) de la tâche d'importation de l'ensemble de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si une tâche d'importation d'un ensemble de données échoue, raison de cet échec.

Type : chaîne

Obligatoire : non

importMode

Mode d'importation utilisé par la tâche d'importation de l'ensemble de données pour mettre à jour les données de l'ensemble de données. Pour plus d'informations, voir Mise à jour des données groupées existantes.

Type : String

Valeurs valides : `FULL | INCREMENTAL`

Obligatoire : non

jobName

Nom de la tâche d'importation de l'ensemble de données.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le statut de la tâche d'importation de l'ensemble de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

status

État de la tâche d'importation de l'ensemble de données.

Une tâche d'importation de jeux de données peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetSchema

Service : Amazon Personalize

Décrit le schéma d'un ensemble de données. Pour plus d'informations sur les schémas, consultez CreateSchema.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles le schéma a été créé.

Type : Timestamp

Obligatoire : non

domain

Le domaine d'un schéma que vous avez créé pour un jeu de données dans un groupe de jeux de données Domaine.

Type : String

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le schéma a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom du schéma.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

schema

Le schéma.

Type : String

Contraintes de longueur : longueur maximale de 20 000.

Obligatoire : non

schemaArn

L'Amazon Resource Name (ARN) du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetSchemaSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'un schéma de jeu de données. Pour obtenir une liste complète, appelez l'DescribeSchemaAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles le schéma a été créé.

Type : Timestamp

Obligatoire : non

domain

Le domaine d'un schéma que vous avez créé pour un jeu de données dans un groupe de jeux de données Domaine.

Type : String

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le schéma a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom du schéma.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

schemaArn

L'Amazon Resource Name (ARN) du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DatasetSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'un ensemble de données. Pour obtenir une liste complète, appelez l'DescribeDatasetAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles le jeu de données a été créé.

Type : Timestamp

Obligatoire : non

datasetArn

L'Amazon Resource Name (ARN) du jeu de données.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

datasetType

Le type de jeu de données. L’une des valeurs suivantes :

- Interactions
- Éléments
- Utilisateurs
- Interactions événementielles

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `^[A-Za-z_]+$`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le jeu de données a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom de l'ensemble de données.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État de l'ensemble de données.

Un jeu de données peut avoir l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

DatasetUpdateSummary

Service : Amazon Personalize

Décrit une mise à jour d'un ensemble de données.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la mise à jour du jeu de données.

Type : Timestamp

Obligatoire : non

failureReason

Si la mise à jour d'un ensemble de données échoue, veuillez en indiquer la raison.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure de la dernière mise à jour (en heure Unix) de l'ensemble de données.

Type : Timestamp

Obligatoire : non

schemaArn

Nom de ressource Amazon (ARN) du schéma qui a remplacé le schéma précédent de l'ensemble de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la mise à jour du jeu de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DataSource

Service : Amazon Personalize

Décrit la source de données qui contient les données à charger dans un ensemble de données ou la liste des enregistrements à supprimer d'Amazon Personalize.

Table des matières

dataLocation

Pour les tâches d'importation de jeux de données, le chemin d'accès au compartiment Amazon S3 où sont stockées les données que vous souhaitez télécharger dans votre ensemble de données. Pour les tâches de suppression de données, chemin d'accès au compartiment Amazon S3 qui stocke la liste des enregistrements à supprimer.

Par exemple :

`s3://bucket-name/folder-name/fileName.csv`

Si vos fichiers CSV se trouvent dans un dossier de votre compartiment Amazon S3 et que vous souhaitez que votre tâche d'importation ou de suppression de données prenne en compte plusieurs fichiers, vous pouvez spécifier le chemin d'accès au dossier. Dans le cadre d'une tâche de suppression de données, Amazon Personalize utilise tous les fichiers du dossier et tous les sous-dossiers. Utilisez la syntaxe suivante avec un / après le nom du dossier :

`s3://bucket-name/folder-name/`

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `(s3|http|https)://.+`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DefaultCategoricalHyperParameterRange

Service : Amazon Personalize

Indique le nom et la plage par défaut d'un hyperparamètre catégorique et indique si l'hyperparamètre est réglable. La valeur d'un hyperparamètre réglable peut être déterminée lors de l'optimisation des hyperparamètres (HPO).

Table des matières

isTunable

Si l'hyperparamètre est réglable.

Type : booléen

Obligatoire : non

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

values

Liste des catégories de l'hyperparamètre.

Type : tableau de chaînes

Membres du tableau : nombre maximum de 100 éléments.

Contraintes de longueur : Longueur maximum de 1 000.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DefaultContinuousHyperParameterRange

Service : Amazon Personalize

Indique le nom et la plage par défaut d'un hyperparamètre continu et indique si l'hyperparamètre est réglable. La valeur d'un hyperparamètre réglable peut être déterminée lors de l'optimisation des hyperparamètres (HPO).

Table des matières

isTunable

Si l'hyperparamètre est réglable.

Type : booléen

Obligatoire : non

maxValue

Valeur maximale autorisée pour l'hyperparamètre.

Type : double

Plage valide : valeur minimale de -1000000.

Obligatoire : non

minValue

Valeur minimale autorisée pour l'hyperparamètre.

Type : double

Plage valide : valeur minimale de -1000000.

Obligatoire : non

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DefaultHyperParameterRanges

Service : Amazon Personalize

Spécifie les hyperparamètres et leurs plages par défaut. Les hyperparamètres peuvent être catégoriques, continus ou à valeurs entières.

Table des matières

categoricalHyperParameterRanges

Les hyperparamètres catégoriels et leurs plages par défaut.

Type : tableau d'objets DefaultCategoricalHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

continuousHyperParameterRanges

Les hyperparamètres continus et leurs plages par défaut.

Type : tableau d'objets DefaultContinuousHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

integerHyperParameterRanges

Les hyperparamètres à valeur entière et leurs plages par défaut.

Type : tableau d'objets DefaultIntegerHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DefaultIntegerHyperParameterRange

Service : Amazon Personalize

Indique le nom et la plage par défaut d'un hyperparamètre à valeur entière et indique si l'hyperparamètre est réglable. La valeur d'un hyperparamètre réglable peut être déterminée lors de l'optimisation des hyperparamètres (HPO).

Table des matières

isTunable

Indique si l'hyperparamètre est réglable.

Type : booléen

Obligatoire : non

maxValue

Valeur maximale autorisée pour l'hyperparamètre.

Type : entier

Plage valide : valeur maximale de 1000000.

Obligatoire : non

minValue

Valeur minimale autorisée pour l'hyperparamètre.

Type : entier

Plage valide : valeur minimale de -1000000.

Obligatoire : non

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventParameters

Service : Amazon Personalize

Décrit les paramètres des événements utilisés lors de la création de solutions.

Table des matières

eventType

Nom du type d'événement à prendre en compte pour la création de la solution.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

eventValueThreshold

Le seuil du type d'événement. Seuls les événements dont la valeur est supérieure ou égale à ce seuil seront pris en compte pour la création de la solution.

Type : double

Obligatoire : non

weight

Le poids du type d'événement. Une pondération plus élevée signifie une plus grande importance du type d'événement pour la solution créée.

Type : double

Plage valide : Valeur minimum de 0. Valeur maximale de 1.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventsConfig

Service : Amazon Personalize

Décrit la configuration des événements utilisés lors de la création de solutions.

Table des matières

eventParametersList

Liste de paramètres d'événements, qui inclut les types d'événements ainsi que leurs seuils et pondérations de valeur d'événement.

Type : tableau d’objets EventParameters

Membres du tableau : nombre maximum de 10 éléments.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventTracker

Service : Amazon Personalize

Fournit des informations sur un outil de suivi des événements.

Table des matières

accountId

Le AWS compte propriétaire de l'outil de suivi des événements.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

creationDateTime

Date et heure (au format Unix) auxquelles le suivi des événements a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de données qui reçoit les données de l'événement.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

eventTrackerArn

L'ARN du suivi des événements.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le suivi des événements a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom de l'outil de suivi des événements.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État du suivi des événements.

Un outil de suivi d'événements peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

trackingId

L'identifiant du suivi des événements. Incluez cet ID dans les demandes adressées à l'PutEventsAPI.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventTrackerSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'un outil de suivi des événements. Pour obtenir une liste complète, appelez l'DescribeEventTrackerAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles le suivi des événements a été créé.

Type : Timestamp

Obligatoire : non

eventTrackerArn

Le nom de ressource Amazon (ARN) du suivi des événements.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles le suivi des événements a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom de l'outil de suivi des événements.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État du suivi des événements.

Un outil de suivi d'événements peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

FeatureTransformation

Service : Amazon Personalize

Fournit des informations sur la transformation des fonctionnalités. La transformation de fonctionnalité est le processus de transformation des données d'entrée brutes en un format plus approprié à la formation du modèle.

Table des matières

creationDateTime

Date et heure de création (en heure Unix) de la transformation de la fonctionnalité.

Type : Timestamp

Obligatoire : non

defaultParameters

Fournit les paramètres par défaut pour la transformation des entités.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1000.

Obligatoire : non

featureTransformationArn

Le nom de ressource Amazon (ARN) de l' FeatureTransformation objet.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Date et heure de la dernière mise à jour (en heure Unix) de la transformation des fonctionnalités.

Type : Timestamp

Obligatoire : non

name

Nom de la transformation de fonction.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État de la transformation des fonctionnalités.

Une transformation de fonction peut se présenter dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

FieldsForThemeGeneration

Service : Amazon Personalize

Un mappage de chaîne à chaîne des détails de configuration pour la génération du thème.

Table des matières

itemName

Nom de la colonne du jeu de données Items qui stocke le nom de chaque élément du jeu de données.

Type : String

Contraintes de longueur : longueur maximale de 150.

Modèle : `[A-Za-z_][A-Za-z\d_]*`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Filter

Service : Amazon Personalize

Contient des informations sur un filtre de recommandation, notamment son ARN, son statut et son expression de filtre.

Table des matières

creationDateTime

Heure à laquelle le filtre a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

L'ARN du groupe de jeux de données auquel appartient le filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si le filtre a échoué, la raison de son échec.

Type : chaîne

Obligatoire : non

filterArn

L'ARN du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterExpression

Spécifie le type d'interactions entre éléments à filtrer dans les résultats des recommandations. L'expression du filtre doit respecter des règles de format spécifiques. Pour plus d'informations sur la structure et la syntaxe des expressions de filtre, consultez la section Expressions de filtre.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 2500.

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle le filtre a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom du filtre.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

FilterSummary

Service : Amazon Personalize

Bref résumé des attributs d'un filtre.

Table des matières

creationDateTime

Heure à laquelle le filtre a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

L'ARN du groupe de jeux de données auquel appartient le filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si le filtre a échoué, raison de l'échec.

Type : chaîne

Obligatoire : non

filterArn

L'ARN du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Heure à laquelle le filtre a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Nom du filtre.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État du filtre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

HPOConfig

Service : Amazon Personalize

Décrit les propriétés de l'optimisation des hyperparamètres (HPO).

Table des matières

algorithmHyperParameterRanges

Hyperparamètres et plages autorisées pour ceux-ci.

Type : objet HyperParameterRanges

Obligatoire : non

hpoObjective

Métrique à optimiser dans le cadre de la procédure HPO.

> **Note**
>
> Pour le moment, Amazon Personalize ne prend pas en charge la configuration de `hpoObjective`.

Type : objet HPOObjective

Obligatoire : non

hpoResourceConfig

Décrit la configuration des ressources pour la procédure HPO.

Type : objet HPOResourceConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

HPOObjective

Service : Amazon Personalize

Métrique à optimiser dans le cadre de la procédure d'optimisation des hyperparamètres (HPO).

> **Note**
>
> Pour le moment, Amazon Personalize ne prend pas en charge la configuration de `hpoObjective`.

Table des matières

metricName

Le nom de la métrique.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

metricRegex

Expression régulière permettant de trouver la métrique dans les journaux des tâches de formation.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

type

Type de la métrique. Les valeurs valides sont `Maximize` et `Minimize`.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

HPOResourceConfig

Service : Amazon Personalize

Décrit la configuration des ressources dans le cadre de la procédure d'optimisation des hyperparamètres (HPO).

Table des matières

maxNumberOfTrainingJobs

Nombre maximal de tâches de formation lorsque vous créez une version de la solution. La valeur maximale de `maxNumberOfTrainingJobs` est `40`.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

maxParallelTrainingJobs

Nombre maximal de tâches de formation parallèles lorsque vous créez une version de la solution. La valeur maximale de `maxParallelTrainingJobs` est `10`.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

HyperParameterRanges

Service : Amazon Personalize

Spécifie les hyperparamètres et leurs plages. Les hyperparamètres peuvent être catégoriques, continus ou à valeurs entières.

Table des matières

categoricalHyperParameterRanges

Les hyperparamètres catégoriques et leurs plages.

Type : tableau d'objets CategoricalHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

continuousHyperParameterRanges

Les hyperparamètres continus et leurs plages.

Type : tableau d'objets ContinuousHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

integerHyperParameterRanges

Les hyperparamètres à valeur entière et leurs plages.

Type : tableau d'objets IntegerHyperParameterRange

Membres du tableau : nombre maximum de 100 éléments.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

IntegerHyperParameterRange

Service : Amazon Personalize

Fournit le nom et la plage d'un hyperparamètre à valeurs entières.

Table des matières

maxValue

Valeur maximale autorisée pour l'hyperparamètre.

Type : entier

Plage valide : valeur maximale de 1000000.

Obligatoire : non

minValue

Valeur minimale autorisée pour l'hyperparamètre.

Type : entier

Plage valide : valeur minimale de -1000000.

Obligatoire : non

name

Nom de l'hyperparamètre.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

MetricAttribute

Service : Amazon Personalize

Contient des informations sur une métrique sur laquelle une attribution de métrique rend compte. Pour plus d'informations, voir Mesurer l'impact des recommandations.

Table des matières

eventType

Type d'événement de la métrique.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

expression

Expression de l'attribut. Les fonctions disponibles sont `SUM()` ou`SAMPLECOUNT()`. Pour les fonctions SUM (), indiquez le type de jeu de données (Interactions ou Items) et la colonne à additionner en tant que paramètre. Par exemple SUM (Items.price).

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

metricName

Le nom de la métrique. Le nom vous aide à identifier la métrique dans Amazon CloudWatch ou Amazon S3.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

MetricAttribution

Service : Amazon Personalize

Contient des informations sur l'attribution d'une métrique. Une attribution métrique crée des rapports sur les données que vous importez dans Amazon Personalize. Selon la manière dont vous importez les données, vous pouvez consulter les rapports dans Amazon CloudWatch ou Amazon S3. Pour plus d'informations, consultez la section Mesurer l'impact des recommandations.

Table des matières

creationDateTime

Date et heure de création de l'attribution métrique.

Type : Timestamp

Obligatoire : non

datasetGroupArn

Le groupe de données Amazon Resource Name (ARN) associé à l'attribution métrique est un groupe de données.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

La raison de l'échec de l'attribution des métriques.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure de la dernière mise à jour de l'attribution métrique.

Type : Timestamp

Obligatoire : non

metricAttributionArn

L'attribution de la métrique est le Amazon Resource Name (ARN).

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

metricsOutputConfig

Configuration de sortie de l'attribution métrique.

Type : objet MetricAttributionOutput

Obligatoire : non

name

Le nom de l'attribution de la métrique.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État de l'attribution des métriques.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

MetricAttributionOutput

Service : Amazon Personalize

Détails de configuration de sortie pour une attribution de métrique.

Table des matières

roleArn

Le nom de ressource Amazon (ARN) du rôle de service IAM autorisé à ajouter des données à votre compartiment Amazon S3 de sortie et à ajouter des métriques à Amazon CloudWatch. Pour plus d'informations, voir Mesurer l'impact des recommandations.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatoire : oui

s3DataDestination

Les détails de configuration d'un compartiment d'entrée ou de sortie Amazon S3.

Type : objet S3DataConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

MetricAttributionSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une attribution métrique. Pour une liste complète, appelez le DescribeMetricAttribution.

Table des matières

creationDateTime

Date et heure de création de l'attribution métrique.

Type : Timestamp

Obligatoire : non

failureReason

La raison de l'échec de l'attribution des métriques.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure de la dernière mise à jour de l'attribution métrique.

Type : Timestamp

Obligatoire : non

metricAttributionArn

L'attribution de la métrique est le Amazon Resource Name (ARN).

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

name

Le nom de l'attribution métrique.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

status

État de l'attribution des métriques.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

OptimizationObjective

Service : Amazon Personalize

Décrit l'objectif supplémentaire de la solution, tel que l'optimisation des minutes de streaming ou l'augmentation des revenus. Pour plus d'informations, voir Optimisation d'une solution.

Table des matières

itemAttribute

La colonne de métadonnées numériques d'un jeu de données Items associée à l'objectif d'optimisation. Par exemple, VIDEO_LENGTH (pour maximiser les minutes de streaming) ou PRICE (pour maximiser les revenus).

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 150

Obligatoire : non

objectiveSensitivity

Spécifie comment Amazon Personalize équilibre l'importance de votre objectif d'optimisation par rapport à la pertinence.

Type : String

Valeurs valides : `LOW | MEDIUM | HIGH | OFF`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Recipe

Service : Amazon Personalize

Fournit des informations sur une recette. Chaque recette fournit un algorithme qu'Amazon Personalize utilise lors de l'entraînement du modèle lorsque vous utilisez l'CreateSolutionopération.

Table des matières

algorithmArn

Amazon Resource Name (ARN) de l'algorithme utilisé par Amazon Personalize pour entraîner le modèle.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

creationDateTime

Date et heure (au format Unix) auxquelles la recette a été créée.

Type : Timestamp

Obligatoire : non

description

Description de la recette.

Type : chaîne

Obligatoire : non

featureTransformationArn

L'ARN de l' FeatureTransformation objet.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Date et heure (au format Unix) auxquelles la recette a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom de la recette.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

recipeType

L'une des valeurs suivantes :

- PERSONALIZED-RANKING
- RELATED_ITEMS
- USER_PERSONALIZATION

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

status

État de la recette.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

RecipeSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une recette. Pour obtenir une liste complète, appelez l'DescribeRecipeAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles la recette a été créée.

Type : Timestamp

Obligatoire : non

domain

Le domaine de la recette (si la recette est un cas d'utilisation d'un groupe de jeux de données de domaine).

Type : String

Valeurs valides : `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la recette a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom de la recette.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la recette.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Recommender

Service : Amazon Personalize

Décrit un générateur de recommandations pour un groupe de jeux de données de domaine. Vous créez un recommandeur dans un groupe de jeux de données de domaine pour un cas d'utilisation de domaine spécifique (recette de domaine), et vous spécifiez le recommandateur dans une GetRecommendationsdemande.

Table des matières

creationDateTime

Date et heure (au format Unix) auxquelles le recommandeur a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de données de domaine qui contient le recommandateur.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

failureReason

Si un système de recommandation échoue, la raison de l'échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (au format Unix) auxquelles le recommandeur a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

latestRecommenderUpdate

Fournit un résumé des dernières mises à jour du logiciel de recommandation.

Type : objet RecommenderUpdateSummary

Obligatoire : non

modelMetrics

Fournit des mesures d'évaluation qui vous aident à déterminer les performances d'un recommandeur. Pour plus d'informations, consultez la section Évaluation d'un recommandeur.

Type : mappage chaîne à double

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Obligatoire : non

name

Le nom du recommandant.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette (cas d'utilisation du groupe de données de domaine) pour laquelle le recommandeur a été créé.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

recommenderArn

Le nom de ressource Amazon (ARN) du recommandeur.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

recommenderConfig

Les détails de configuration du recommandeur.

Type : objet RecommenderConfig

Obligatoire : non

status

Le statut du recommandant.

Un recommandeur peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- ARRÊT EN ATTENTE > ARRÊT EN COURS > INACTIF > DÉBUT EN ATTENTE > DÉMARRAGE EN COURS > ACTIF
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

RecommenderConfig

Service : Amazon Personalize

Les détails de configuration du recommandeur.

Table des matières

enableMetadataWithRecommendations

Si les métadonnées contenant des recommandations sont activées pour le recommandeur. Si cette option est activée, vous pouvez spécifier les colonnes de votre jeu de données Items dans votre demande de recommandations. Amazon Personalize renvoie ces données pour chaque article dans la réponse de recommandation. Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultez la section Activation des métadonnées dans les recommandations pour un recommandeur.

Si vous activez les métadonnées dans les recommandations, vous devrez payer des frais supplémentaires. Pour plus d'informations, consultez les tarifs d'Amazon Personalize.

Type : booléen

Obligatoire : non

itemExplorationConfig

Spécifie les hyperparamètres de configuration d'exploration, notamment `explorationWeight` et `explorationItemAgeCutOff` que vous souhaitez utiliser pour configurer la quantité d'objets explorés par Amazon Personalize lorsqu'il recommande des articles. Fournissez `itemExplorationConfig` des données uniquement si vos recommandeurs génèrent des recommandations personnalisées pour un utilisateur (articles peu populaires ou articles similaires).

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

minRecommendationRequestsPerSecond

Spécifie le nombre minimum de demandes de recommandation provisionnées par seconde qu'Amazon Personalize prendra en charge. Un montant élevé `minRecommendationRequestsPerSecond` fera augmenter votre facture. Nous vous recommandons de commencer par 1 pour `minRecommendationRequestsPerSecond` (valeur par défaut). Suivez votre consommation à l'aide CloudWatch des statistiques Amazon et `minRecommendationRequestsPerSecond` augmentez-la si nécessaire.

Type : entier

Plage valide : Valeur minimum de 1.

Obligatoire : non

trainingDataConfig

Spécifie la configuration des données de formation à utiliser lors de la création d'un recommandeur de domaine.

Type : objet TrainingDataConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

RecommenderSummary

Service : Amazon Personalize

Fournit un résumé des propriétés du recommandeur.

Table des matières

creationDateTime

Date et heure (au format Unix) auxquelles le recommandeur a été créé.

Type : Timestamp

Obligatoire : non

datasetGroupArn

Le nom de ressource Amazon (ARN) du groupe de données de domaine qui contient le recommandateur.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

lastUpdatedDateTime

Date et heure (au format Unix) auxquelles le recommandeur a été mis à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

name

Le nom du recommandant.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette (cas d'utilisation du groupe de données de domaine) pour laquelle le recommandeur a été créé.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

recommenderArn

Le nom de ressource Amazon (ARN) du recommandeur.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

recommenderConfig

Les détails de configuration du recommandeur.

Type : objet RecommenderConfig

Obligatoire : non

status

Le statut du recommandant. Un recommandeur peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- ARRÊT EN ATTENTE > ARRÊT EN COURS > INACTIF > DÉBUT EN ATTENTE > DÉMARRAGE EN COURS > ACTIF
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

RecommenderUpdateSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une mise à jour de recommandation. Pour obtenir une liste complète, appelez l'DescribeRecommenderAPI.

Table des matières

creationDateTime

Date et heure (au format Unix) auxquelles la mise à jour du recommandateur a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si la mise à jour d'une recommandation échoue, raison de l'échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la dernière mise à jour du recommandeur a été mise à jour.

Type : Timestamp

Obligatoire : non

recommenderConfig

Les détails de configuration de la mise à jour du recommandateur.

Type : objet RecommenderConfig

Obligatoire : non

status

État de la mise à jour du recommandateur. La mise à jour d'une recommandation peut se présenter dans l'un des états suivants :

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

S3DataConfig

Service : Amazon Personalize

Les détails de configuration d'un compartiment d'entrée ou de sortie Amazon S3.

Table des matières

path

Le chemin du fichier du compartiment Amazon S3.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `(s3|http|https)://.+`

Obligatoire : oui

kmsKeyArn

Le nom de ressource Amazon (ARN) de la clé AWS Key Management Service (KMS) utilisée par Amazon Personalize pour chiffrer ou déchiffrer les fichiers d'entrée et de sortie.

Type : String

Contraintes de longueur : longueur maximale de 2048.

Modèle : `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Solution

Service : Amazon Personalize

> **Important**
>
> Par défaut, toutes les nouvelles solutions utilisent l'entraînement automatique. Avec la formation automatique, vous encourez des frais de formation pendant que votre solution est active. Pour éviter des coûts inutiles, lorsque vous avez terminé, vous pouvez mettre à jour la solution pour désactiver l'entraînement automatique. Pour plus d'informations sur les coûts de formation, consultez la tarification d'Amazon Personalize.

Un objet qui fournit des informations sur une solution. Une solution inclut la recette personnalisée, les paramètres personnalisés et les modèles entraînés (versions de solution) qu'Amazon Personalize utilise pour générer des recommandations.

Une fois que vous avez créé une solution, vous ne pouvez pas modifier sa configuration. Si vous devez apporter des modifications, vous pouvez cloner la solution avec la console Amazon Personalize ou en créer une nouvelle.

Table des matières

autoMLResult

Lorsque `performAutoML` c'est vrai, indique la meilleure recette trouvée.

Type : objet AutoMLResult

Obligatoire : non

creationDateTime

Date et heure de création (en heure Unix) de la solution.

Type : Timestamp

Obligatoire : non

datasetGroupArn

L'Amazon Resource Name (ARN) du groupe de jeux de données qui fournit les données d'entraînement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

eventType

Le type d'événement (par exemple, « clic » ou « mention J'aime ») qui est utilisé pour entraîner le modèle. Si aucun `eventType` n'est fourni, Amazon Personalize utilise toutes les interactions pour l'entraînement avec la même pondération, quel que soit le type.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) de la dernière mise à jour de la solution.

Type : Timestamp

Obligatoire : non

latestSolutionUpdate

Fournit un résumé des dernières mises à jour de la solution.

Type : objet SolutionUpdateSummary

Obligatoire : non

latestSolutionVersion

Décrit la dernière version de la solution, y compris le statut et l'ARN.

Type : objet SolutionVersionSummary

Obligatoire : non

name

Nom de la solution.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

performAutoML

> **Important**
>
> Nous vous déconseillons d'activer le machine learning. Adaptez plutôt les recettes Amazon Personalize disponibles à votre cas d'utilisation. Pour plus d'informations, consultez Détermination de votre cas d'utilisation.

Lorsque ce paramètre a la valeur true (vrai), Amazon Personalize recherche la meilleure recette USER_PERSONALIZATION dans la liste spécifiée dans la configuration de la solution (`recipeArn` ne doit pas être précisé). Lorsque ce paramètre a la valeur false (valeur par défaut), Amazon Personalize utilise `recipeArn` pour l'entraînement.

Type : booléen

Obligatoire : non

performAutoTraining

Spécifie si la solution crée automatiquement des versions de solution. La valeur par défaut est `True` et la solution crée automatiquement de nouvelles versions de solution tous les 7 jours.

Pour plus d'informations sur l'entraînement automatique, consultez la section Création et configuration d'une solution.

Type : booléen

Obligatoire : non

performHPO

Indique s'il convient d'effectuer une optimisation des hyperparamètres sur la recette choisie. La valeur par défaut est `false`.

Type : booléen

Obligatoire : non

performIncrementalUpdate

Valeur booléenne qui indique si des mises à jour d'entraînement incrémentielles sont effectuées sur le modèle. Lorsqu'il est activé, cela permet au modèle d'apprendre plus fréquemment à partir de nouvelles données sans nécessiter de réentraînement complet, ce qui permet une personnalisation en temps quasi réel. Ce paramètre n'est pris en charge que pour les solutions qui utilisent la recette de similarité sémantique

Type : booléen

Obligatoire : non

recipeArn

L'ARN de la recette utilisée pour créer la solution. Ceci est obligatoire lorsque `performAutoML` c'est faux.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionArn

L'ARN de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionConfig

Décrit les propriétés de configuration de la solution.

Type : objet SolutionConfig

Obligatoire : non

status

État de la solution.

Une solution peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

SolutionConfig

Service : Amazon Personalize

Décrit les propriétés de configuration de la solution.

Table des matières

algorithmHyperParameters

Répertorie les hyperparamètres de l'algorithme et leurs valeurs.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

autoMLConfig

L'MLConfigobjet Auto contenant une liste de recettes à rechercher lorsque AutoML est exécuté.

Type : objet AutoMLConfig

Obligatoire : non

autoTrainingConfig

Spécifie la configuration d'entraînement automatique à utiliser.

Type : objet AutoTrainingConfig

Obligatoire : non

eventsConfig

Décrit la configuration d'un événement, qui inclut une liste de paramètres d'événement. Vous pouvez définir jusqu'à 10 paramètres d'événement. Les événements sont utilisés dans la création de solutions.

Type : objet EventsConfig

Obligatoire : non

eventValueThreshold

Seuls les événements dont la valeur est supérieure ou égale à ce seuil sont utilisés pour l'entraînement d'un modèle.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

featureTransformationParameters

Répertorie les paramètres de transformation de fonctionnalité.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

hpoConfig

Décrit les propriétés de l'optimisation des hyperparamètres (HPO).

Type : objet HPOConfig

Obligatoire : non

optimizationObjective

Décrit l'objectif supplémentaire de la solution, tel que l'optimisation des minutes de streaming ou l'augmentation des revenus. Pour plus d'informations, voir Optimisation d'une solution.

Type : objet OptimizationObjective

Obligatoire : non

trainingDataConfig

Spécifie la configuration des données d'entraînement à utiliser lors de la création d'une version de solution personnalisée (modèle entraîné).

Type : objet TrainingDataConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

SolutionSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une solution. Pour obtenir une liste complète, appelez l'DescribeSolutionAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles la solution a été créée.

Type : Timestamp

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) de la dernière mise à jour de la solution.

Type : Timestamp

Obligatoire : non

name

Nom de la solution.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

recipeArn

Le nom de ressource Amazon (ARN) de la recette utilisée par la solution.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionArn

L'Amazon Resource Name (ARN) de la solution.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la solution.

Une solution peut se trouver dans l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

SolutionUpdateConfig

Service : Amazon Personalize

Les détails de configuration de la mise à jour de la solution.

Table des matières

autoTrainingConfig

La configuration d'entraînement automatique à utiliser lorsque `performAutoTraining` c'est vrai.

Type : objet AutoTrainingConfig

Obligatoire : non

eventsConfig

Décrit la configuration d'un événement, qui inclut une liste de paramètres d'événement. Vous pouvez définir jusqu'à 10 paramètres d'événement. Les événements sont utilisés dans la création de solutions.

Type : objet EventsConfig

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

SolutionUpdateSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une mise à jour de solution. Pour obtenir une liste complète, appelez l'DescribeSolutionAPI.

Table des matières

creationDateTime

Date et heure (au format Unix) auxquelles la mise à jour de la solution a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si la mise à jour d'une solution échoue, raison de l'échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la dernière mise à jour de la solution a été mise à jour.

Type : Timestamp

Obligatoire : non

performAutoTraining

Si la solution crée automatiquement des versions de solution.

Type : booléen

Obligatoire : non

performIncrementalUpdate

Valeur booléenne qui indique si des mises à jour d'entraînement incrémentielles sont effectuées sur le modèle. Lorsque cette option est activée, une formation progressive est effectuée pour les versions de solution comportant des campagnes actives et permet au modèle d'apprendre

plus fréquemment à partir de nouvelles données sans nécessiter de formation complémentaire complète, ce qui permet une personnalisation en temps quasi réel. Ce paramètre n'est pris en charge que pour les solutions qui utilisent la recette de similarité sémantique. La valeur par défaut est `true`.

Notez que certains scores et attributs (tels que les mises à jour de la popularité des objets et de leur fraîcheur utilisées pour influencer le classement avec la aws-semantic-similarity recette) peuvent ne pas être mis à jour avant le prochain entraînement complet.

Type : booléen

Obligatoire : non

solutionUpdateConfig

Les détails de configuration de la solution.

Type : objet SolutionUpdateConfig

Obligatoire : non

status

État de la mise à jour de la solution. Une mise à jour de solution peut se présenter dans l'un des états suivants :

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

SolutionVersion

Service : Amazon Personalize

Objet qui fournit des informations sur une version spécifique d'une solution dans un groupe de jeux de données personnalisé.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles cette version de la solution a été créée.

Type : Timestamp

Obligatoire : non

datasetGroupArn

Nom de ressource Amazon (ARN) du groupe de données fournissant les données d'entraînement.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

eventType

Le type d'événement (par exemple, « clic » ou « mention J'aime ») qui est utilisé pour entraîner le modèle.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

failureReason

Si la formation d'une version de solution échoue, raison de l'échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) de la dernière mise à jour de la solution.

Type : Timestamp

Obligatoire : non

name

Nom de la version de la solution.

Type : Chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

performAutoML

Lorsque cela est vrai, Amazon Personalize recherche la recette la plus optimale en fonction de la configuration de la solution. Lorsque la valeur est False (valeur par défaut), Amazon Personalize utilise`recipeArn`.

Type : booléen

Obligatoire : non

performHPO

Indique s'il convient d'effectuer une optimisation des hyperparamètres sur la recette choisie. La valeur par défaut est `false`.

Type : booléen

Obligatoire : non

performIncrementalUpdate

Si la version de la solution doit effectuer une mise à jour incrémentielle. Lorsqu'il est défini sur true, l'entraînement ne traitera que les données modifiées depuis le dernier entraînement, comme lorsque TrainingMode est défini sur UPDATE. Cela ne peut être utilisé qu'avec les versions de solution qui utilisent la recette de personnalisation utilisateur.

Type : booléen

Obligatoire : non

recipeArn

L'ARN de la recette utilisée dans la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionArn

L'ARN de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

solutionConfig

Décrit les propriétés de configuration de la solution.

Type : objet SolutionConfig

Obligatoire : non

solutionVersionArn

L'ARN de la version de la solution.

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la version de la solution.

Une version de solution peut présenter l'un des états suivants :

- CRÉER EN ATTENTE
- CRÉER IN_PROGRESS
- ACTIF
- ÉCHEC DE LA CRÉATION
- CRÉER UN ARRÊT
- CRÉATION ARRÊTÉE

Type : Chaîne

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

trainingHours

Le temps utilisé pour entraîner le modèle. Le temps nécessaire à l'entraînement d'un mannequin vous est facturé. Ce champ n'est visible qu'une fois qu'Amazon Personalize a correctement entraîné un modèle.

Type : double

Plage valide : Valeur minimum de 0.

Obligatoire : non

trainingMode

L'étendue de la formation à effectuer lors de la création de la version de la solution. Une `FULL` formation prend en compte toutes les données de votre groupe de jeux de données. An `UPDATE` traite uniquement les données modifiées depuis la dernière formation. Seules les versions de solution créées avec la recette de personnalisation utilisateur peuvent être utilisées. `UPDATE`

Type : Chaîne

Valeurs valides : `FULL | UPDATE | AUTOTRAIN`

Obligatoire : non

trainingType

Si la version de la solution a été créée automatiquement ou manuellement.

Type : Chaîne

Valeurs valides : `AUTOMATIC | MANUAL`

Obligatoire : non

tunedHPOParams

Si l'optimisation des hyperparamètres a été réalisée, contient les valeurs des hyperparamètres du modèle le plus performant.

Type : objet TunedHPOParams

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

SolutionVersionSummary

Service : Amazon Personalize

Fournit un résumé des propriétés d'une version de solution. Pour obtenir une liste complète, appelez l'DescribeSolutionVersionAPI.

Table des matières

creationDateTime

Date et heure (en heure Unix) auxquelles cette version d'une solution a été créée.

Type : Timestamp

Obligatoire : non

failureReason

Si une version de solution échoue, raison de l'échec.

Type : chaîne

Obligatoire : non

lastUpdatedDateTime

Date et heure (en heure Unix) auxquelles la version de la solution a été mise à jour pour la dernière fois.

Type : Timestamp

Obligatoire : non

solutionVersionArn

Le nom de ressource Amazon (ARN) de la version de la solution.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

status

État de la version de la solution.

Une version de solution peut présenter l'un des états suivants :

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -ou- CREATE FAILED

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

trainingMode

L'étendue de la formation à effectuer lors de la création de la version de la solution. Une `FULL` formation prend en compte toutes les données de votre groupe de jeux de données. An ne `UPDATE` traite que les données modifiées depuis la dernière formation. Seules les versions de solution créées avec la recette de personnalisation utilisateur peuvent être utilisées. `UPDATE`

Type : String

Valeurs valides : `FULL | UPDATE | AUTOTRAIN`

Obligatoire : non

trainingType

Si la version de la solution a été créée automatiquement ou manuellement.

Type : String

Valeurs valides : `AUTOMATIC | MANUAL`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

Tag

Service : Amazon Personalize

Les métadonnées facultatives que vous appliquez aux ressources pour vous aider à les classer et à les organiser. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Pour plus d'informations, consultez Tagging Amazon Personalize resources.

Table des matières

tagKey

Partie d'une paire clé-valeur constituant une balise. Une clé est une étiquette générale qui fait office de catégorie pour les valeurs d'étiquette plus spécifiques.

Type : String

Contraintes de longueur : Longueur minimum de 1. Longueur maximale de 128.

Modèle : `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatoire : oui

tagValue

Partie facultative d'une paire clé-valeur qui constitue une balise. Une valeur agit comme un descripteur au sein d'une catégorie d'étiquette (clé).

Type : String

Contraintes de longueur : Longueur minimum de 0. Longueur maximum de 256.

Modèle : `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

ThemeGenerationConfig

Service : Amazon Personalize

Les détails de configuration pour générer des thèmes à l'aide d'une tâche d'inférence par lots.

Table des matières

fieldsForThemeGeneration

Champs utilisés pour générer des thèmes descriptifs pour une tâche d'inférence par lots.

Type : objet FieldsForThemeGeneration

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

TrainingDataConfig

Service : Amazon Personalize

Configuration des données de formation à utiliser lors de la création d'un recommandeur de domaine ou d'une version de solution personnalisée (modèle entraîné).

Table des matières

excludedDatasetColumns

Spécifie les colonnes à exclure de l'entraînement. Chaque clé est un type de jeu de données et chaque valeur est une liste de colonnes. Excluez les colonnes pour contrôler les données utilisées par Amazon Personalize pour générer des recommandations.

Par exemple, vous pouvez avoir une colonne que vous souhaitez utiliser uniquement pour filtrer les recommandations. Vous pouvez exclure cette colonne de la formation et Amazon Personalize ne la prend en compte que lors du filtrage.

TYpe : mappage de chaîne à tableau de chaînes

Entrées cartographiques : nombre maximum de 3 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Modèle de clé :`^[A-Za-z_]+$`

Membres du tableau : nombre maximum de 50 éléments.

Contraintes de longueur : longueur maximale de 150.

Modèle : `[A-Za-z_][A-Za-z\d_]*`

Obligatoire : non

includedDatasetColumns

Spécifie les colonnes à inclure à partir de l'entraînement. Chaque clé est un type de jeu de données et chaque valeur est une liste de colonnes. Incluez des colonnes pour contrôler les données utilisées par Amazon Personalize pour générer des recommandations.

Par exemple, vous pouvez avoir plusieurs colonnes dans votre jeu de données d'éléments, mais vous souhaitez utiliser uniquement le titre et la description pour la formation. Vous ne pouvez

inclure que ces deux colonnes pour la formation et Amazon Personalize ne prendra en compte que les colonnes incluses pour la formation.

TYpe : mappage de chaîne à tableau de chaînes

Entrées cartographiques : nombre maximum de 3 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Modèle de clé :`^[A-Za-z_]+$`

Membres du tableau : nombre maximum de 50 éléments.

Contraintes de longueur : longueur maximale de 150.

Modèle : `[A-Za-z_][A-Za-z\d_]*`

Obligatoire : non

Voir aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWSSDKs, consultez ce qui suit :

- AWSSDK pour C++
- AWSSDK pour Java V2
- AWSSDK pour Ruby V3

TunedHPOParams

Service : Amazon Personalize

Si l'optimisation des hyperparamètres (HPO) a été réalisée, contient les valeurs des hyperparamètres du modèle le plus performant.

Table des matières

algorithmHyperParameters

Liste des valeurs d'hyperparamètres du modèle le plus performant.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 100 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 256.

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Amazon Personalize Events

Les types de données suivants sont pris en charge par Amazon Personalize Events :

- Action
- ActionInteraction
- Event
- Item

- MetricAttribution
- User

Action

Service : Amazon Personalize Events

Représente les métadonnées d'action ajoutées à un jeu de données Action à l'aide de l'`PutActions`API. Pour plus d'informations, voir Importer des actions individuellement.

Table des matières

actionId

L'ID associé à l'action.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

properties

Une carte de chaînes de métadonnées spécifiques à une action. Chaque élément de la carte est constitué d'une paire clé-valeur. Par exemple, `{"value": "100"}`.

Les clés utilisent des noms de camel qui correspondent aux champs du schéma du jeu de données Actions. Dans l'exemple précédent, le champ « VALEUR » `value` correspond au champ « VALEUR » défini dans le schéma Actions. Pour les données de chaîne catégorielles, afin d'inclure plusieurs catégories pour une seule action, séparez chaque catégorie par un séparateur de tubes (`|`). Par exemple, `\"Deluxe|Premium\"`.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 32 000

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

ActionInteraction

Service : Amazon Personalize Events

Représente un événement d'interaction d'action envoyé à l'aide de l'`PutActionInteractions`API.

Table des matières

actionId

ID de l'action avec laquelle l'utilisateur a interagi. Cela correspond au `ACTION_ID` champ du schéma d'interaction Action.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

eventType

Type d'événement d'interaction action-action. Vous pouvez spécifier `ViewedTaken`, et les types `Not Taken` d'événements. Pour plus d'informations sur les données de type d'événement d'interaction action-action, consultez la section Données de type d'événement.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

sessionId

L'identifiant associé à la visite de l'utilisateur. Votre application génère un code unique `sessionId` lorsqu'un utilisateur visite votre site Web ou utilise votre application pour la première fois.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

timestamp

Horodatage indiquant le moment où l'événement d'interaction avec l'action s'est produit. Les horodatages doivent être au format Unix Epoch Time, en secondes.

Type : Timestamp

Obligatoire : oui

eventId

Un identifiant associé à l'événement. Si aucun identifiant d'événement n'est fourni, Amazon Personalize génère un identifiant unique pour l'événement. Aucun identifiant d'événement n'est utilisé comme entrée dans le modèle. Amazon Personalize utilise l'identifiant de l'événement pour distinguer les événements uniques. Les événements suivants après le premier avec le même ID d'événement ne sont pas utilisés dans l'entraînement du modèle.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

impression

Liste d'actions IDs qui représente la séquence d'actions que vous avez montrée à l'utilisateur. Par exemple, `["actionId1", "actionId2", "actionId3"]`. Amazon Personalize n'utilise pas les données d'impressions issues d'événements d'interaction liés à l'action. Enregistrez plutôt plusieurs événements pour chaque action et utilisez le type `Viewed` d'événement.

Type : tableau de chaînes

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 25 articles.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

properties

Une carte de chaînes de données spécifiques à un événement que vous pouvez choisir d'enregistrer. Par exemple, si un utilisateur effectue une action, autre que l'identifiant de l'action, vous pouvez également envoyer le nombre d'actions entreprises par l'utilisateur.

Chaque élément de la carte est constitué d'une paire clé-valeur. Par exemple,

```
{"numberOfActions": "12"}
```

Les clés utilisent des noms de camel qui correspondent aux champs du schéma Action Interactions. Dans l'exemple ci-dessus, `numberOfActions` cela correspondrait au champ « NUMBER_OF_ACTIONS » défini dans le schéma des interactions Action.

Le texte suivant ne peut pas être inclus comme mot clé pour les propriétés (sans distinction majuscules/minuscules).

- userId
- sessionId
- eventType
- timestamp
- ID de recommandation
- impression

Type : String

Contraintes de longueur : Longueur minimum de 1. Longueur maximale de 1024.

Obligatoire : non

recommendationId

ID de la liste de recommandations contenant l'action avec laquelle l'utilisateur a interagi.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 40

Obligatoire : non

userId

ID de l'utilisateur qui a interagi avec l'action. Cela correspond au `USER_ID` champ du schéma d'interaction Action.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Event

Service : Amazon Personalize Events

Représente les informations relatives aux événements d'interaction entre articles envoyées via l'`PutEvents`API.

Table des matières

eventType

Type d'événement, tel qu'un clic ou un téléchargement. Cette propriété correspond au `EVENT_TYPE` champ du schéma de votre jeu de données d'interactions entre articles et dépend des types d'événements que vous suivez.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

sentAt

L'horodatage (en heure Unix) côté client lorsque l'événement s'est produit.

Type : Timestamp

Obligatoire : oui

eventId

Un identifiant associé à l'événement. Si aucun identifiant d'événement n'est fourni, Amazon Personalize génère un identifiant unique pour l'événement. Aucun identifiant d'événement n'est utilisé comme entrée dans le modèle. Amazon Personalize utilise l'identifiant de l'événement pour distinguer les événements uniques. Les événements suivants après le premier avec le même ID d'événement ne sont pas utilisés dans l'entraînement du modèle.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

eventValue

La valeur de l'événement qui correspond au `EVENT_VALUE` champ du schéma des interactions entre articles.

Type : flottant

Obligatoire : non

impression

Liste d'éléments IDs qui représente la séquence d'éléments que vous avez montrés à l'utilisateur. Par exemple, `["itemId1", "itemId2", "itemId3"]`. Fournissez une liste d'éléments pour enregistrer manuellement les données d'impressions pour un événement. Pour plus d'informations sur l'enregistrement des données d'impressions, consultez la section Enregistrement des données d'impressions.

Type : tableau de chaînes

Membres du tableau : Nombre minimum de 1 élément. Nombre maximum de 25 articles.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

itemId

La clé d'ID d'élément qui correspond au `ITEM_ID` champ du schéma du jeu de données d'interactions entre articles.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

metricAttribution

Contient des informations sur l'attribution des métriques associées à un événement. Pour plus d'informations sur les attributions métriques, voir Mesurer l'impact des recommandations.

Type : objet MetricAttribution

Obligatoire : non

properties

Une carte de chaînes de données spécifiques à un événement que vous pouvez choisir d'enregistrer. Par exemple, si un utilisateur évalue un film sur votre site, autre que l'identifiant

du film (`itemId`) et le classement (`eventValue`), vous pouvez également envoyer le nombre d'évaluations de films effectuées par l'utilisateur.

Chaque élément de la carte est constitué d'une paire clé-valeur. Par exemple,

```
{"numberOfRatings": "12"}
```

Les clés utilisent des noms de camel qui correspondent aux champs du schéma du jeu de données Item Interactions. Dans l'exemple ci-dessus, le champ « NUMBER_OF_RATINGS » défini dans le schéma du jeu de données d'interactions Item `numberOfRatings` correspondrait.

Le texte suivant ne peut pas être inclus comme mot clé pour les propriétés (sans distinction majuscules/minuscules).

- userId
- sessionId
- eventType
- timestamp
- ID de recommandation
- impression

Type : String

Contraintes de longueur : Longueur minimum de 1. Longueur maximale de 1024.

Obligatoire : non

recommendationId

ID de la liste de recommandations contenant l'élément avec lequel l'utilisateur a interagi. Fournissez un `recommendationId` pour qu'Amazon Personalize enregistre implicitement les recommandations que vous présentez à votre utilisateur sous forme de données d'impressions. Ou fournissez un `recommendationId` si vous utilisez une attribution métrique pour mesurer l'impact des recommandations.

Pour plus d'informations sur l'enregistrement des données d'impressions, consultez la section Enregistrement des données d'impressions. Pour plus d'informations sur la création d'une attribution métrique, voir Mesurer l'impact des recommandations.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 40

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Item

Service : Amazon Personalize Events

Représente les métadonnées des éléments ajoutées à un jeu de données Items à l'aide de l'`PutItems`API. Pour plus d'informations, voir Importation d'articles individuellement.

Table des matières

itemId

L'ID associé à l'article.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

properties

Une carte de chaînes de métadonnées spécifiques à un élément. Chaque élément de la carte est constitué d'une paire clé-valeur. Par exemple, `{"numberOfRatings": "12"}`.

Les clés utilisent des noms de camel qui correspondent aux champs du schéma du jeu de données Items. Dans l'exemple précédent, `numberOfRatings` correspond au champ « NUMBER_OF_RATINGS » défini dans le schéma Items. Pour les données de chaîne catégorielles, afin d'inclure plusieurs catégories pour un seul élément, séparez chaque catégorie par un séparateur de tubes (`|`). Par exemple, `\"Horror|Action\"`.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 32 000.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

MetricAttribution

Service : Amazon Personalize Events

Contient des informations sur l'attribution d'une métrique associée à un événement. Pour plus d'informations sur les attributions métriques, voir Mesurer l'impact des recommandations.

Table des matières

eventAttributionSource

La source de l'événement, telle qu'un tiers.

Type : String

Contraintes de longueur : longueur maximum de 1 024.

Modèle : `^[\x20-\x7E]*[\x21-\x7E]+[\x20-\x7E]*$`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

User

Service : Amazon Personalize Events

Représente les métadonnées utilisateur ajoutées à un ensemble de données Users à l'aide de l'`PutUsers`API. Pour plus d'informations, voir Importer des utilisateurs individuellement.

Table des matières

userId

L'ID associé à l'utilisateur.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

properties

Une carte de chaînes de métadonnées spécifiques à l'utilisateur. Chaque élément de la carte est constitué d'une paire clé-valeur. Par exemple, `{"numberOfVideosWatched": "45"}`.

Les clés utilisent des noms de camel qui correspondent aux champs du schéma du jeu de données Users. Dans l'exemple précédent, `numberOfVideosWatched` correspond au champ « NUMBER_OF_VIDEOS_WATCHED » défini dans le schéma Users. Pour les données de chaîne catégorielles, afin d'inclure plusieurs catégories pour un seul utilisateur, séparez chaque catégorie par un séparateur de tubes (`|`). Par exemple, `\"Member|Frequent shopper\"`.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 24 000

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

Amazon Personalize Runtime

Les types de données suivants sont pris en charge par Amazon Personalize Runtime :

- PredictedAction
- PredictedItem
- Promotion

PredictedAction

Service : Amazon Personalize Runtime

Objet qui identifie une action.

L'GetActionRecommendationsAPI renvoie une liste de `PredictedAction` s.

Table des matières

actionId

ID de l'action recommandée.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

score

Le score de l'action recommandée. Pour plus d'informations sur les scores d'action, voir Fonctionnement de la notation des recommandations d'action.

Type : double

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

PredictedItem

Service : Amazon Personalize Runtime

Un objet qui identifie un élément.

Les GetRecommendations et GetPersonalizedRanking APIs renvoient une liste de `PredictedItem` s.

Table des matières

itemId

L'identifiant de l'article recommandé.

Type : String

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

metadata

Les métadonnées relatives à l'article proviennent de votre jeu de données Items.

Type : mappage chaîne/chaîne

Contraintes relatives à la longueur des clés : longueur maximale de 150.

Contraintes de longueur de valeur : longueur maximale de 20 000.

Obligatoire : non

promotionName

Nom de la promotion qui incluait l'article prévu.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

reason

Si vous utilisez User-Personalization-v 2, une liste des raisons pour lesquelles l'élément a été inclus dans les recommandations. Les raisons possibles sont les suivantes :

- Article sponsorisé : indique que l'article a été inclus dans le cadre d'une promotion que vous avez appliquée dans votre demande de recommandation.
- Exploration : indique que l'objet a été inclus dans l'exploration. Dans le cadre de l'exploration, les recommandations incluent des éléments présentant moins de données d'interactions ou de pertinence pour l'utilisateur. Pour plus d'informations sur l'exploration, voir Exploration.
- Article populaire - Indique que l'article a été inclus en tant qu'objet populaire de remplacement. Si vous utilisez un filtre, en fonction du nombre de recommandations qu'il supprime, Amazon Personalize peut ajouter des éléments réservés `numResults` pour répondre à votre demande de recommandation. Ces éléments sont des éléments populaires, basés sur les données d'interactions, qui répondent à vos critères de filtre. Ils n'ont pas de score de pertinence pour l'utilisateur.

Type : tableau de chaînes

Contraintes de longueur : longueur maximale de 256.

Obligatoire : non

score

Représentation numérique de la certitude du modèle que l'article sera le prochain élément sélectionné par l'utilisateur. Pour plus d'informations sur la logique de notation, voirScores de recommandation.

Type : double

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

Promotion

Service : Amazon Personalize Runtime

Contient des informations sur une promotion. Une promotion définit des règles commerciales supplémentaires qui s'appliquent à un sous-ensemble configurable d'articles recommandés.

Table des matières

filterArn

Le nom de ressource Amazon (ARN) du filtre utilisé par la promotion. Ce filtre définit les critères relatifs aux articles promus. Pour plus d'informations, consultez la section Filtres de promotion.

Type : String

Contraintes de longueur : longueur maximale de 256.

Modèle : `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatoire : non

filterValues

Les valeurs à utiliser lors de la promotion d'articles. Pour chaque paramètre d'espace réservé dans l'expression de filtre de votre promotion, indiquez le nom du paramètre (le cas échéant) sous forme de clé et la ou les valeurs du filtre comme valeur correspondante. Séparez les valeurs multiples d'un paramètre par une virgule.

Pour les expressions de filtre qui utilisent un `INCLUDE` élément pour inclure des éléments, vous devez fournir des valeurs pour tous les paramètres définis dans l'expression. Pour les filtres contenant des expressions qui utilisent un `EXCLUDE` élément pour exclure des éléments, vous pouvez omettre le`filter-values`. Dans ce cas, Amazon Personalize n'utilise pas cette partie de l'expression pour filtrer les recommandations.

Pour plus d'informations sur la création de filtres, consultez la section Recommandations de filtrage et segments d'utilisateurs.

Type : mappage chaîne/chaîne

Entrées cartographiques : nombre maximum de 25 éléments.

Contraintes relatives à la longueur des clés : longueur maximale de 50.

Modèle de clé :`[A-Za-z0-9_]+`

Contraintes de longueur de valeur : longueur maximale de 1 000.

Obligatoire : non

name

Le nom de la promotion.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Modèle : `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatoire : non

percentPromotedItems

Pourcentage d'articles recommandés auxquels appliquer la promotion.

Type : entier

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des langages spécifiques AWS SDKs, consultez ce qui suit :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Types d'erreurs courants

Cette section répertorie les types d'erreurs courants que ce AWS service peut renvoyer. Tous les services ne renvoient pas tous les types d'erreur répertoriés ici. Pour les erreurs spécifiques à une action d'API pour ce service, consultez la rubrique pour cette action d'API.

AccessDeniedException

Vous n'êtes pas autorisé à effectuer cette action. Vérifiez que votre politique IAM inclut les autorisations requises.

Code d’état HTTP : 403

ExpiredTokenException

Le jeton de sécurité inclus dans la demande a expiré. Demandez un nouveau jeton de sécurité et réessayez.

Code d’état HTTP : 403

IncompleteSignature

La signature de la demande n'est pas conforme aux AWS normes. Vérifiez que vous utilisez des AWS informations d'identification valides et que votre demande est correctement formatée. Si vous utilisez un SDK, assurez-vous qu'il est à jour.

Code d’état HTTP : 403

InternalFailure

La demande ne peut pas être traitée pour le moment en raison d'un problème interne au serveur. Réessayez ultérieurement. Si le problème persiste, contactez le AWS Support.

Code d’état HTTP : 500

MalformedHttpRequestException

Le corps de la demande ne peut pas être traité. Cela se produit généralement lorsque le corps de la demande ne peut pas être décompressé à l'aide de l'algorithme de codage de contenu spécifié. Vérifiez que l'en-tête de codage du contenu correspond au format de compression utilisé.

Code d’état HTTP : 400

NotAuthorized

Vous n'êtes pas autorisé à effectuer cette action. Vérifiez que votre politique IAM inclut les autorisations requises.

Code d'état HTTP : 401

OptInRequired

Votre AWS compte nécessite un abonnement pour ce service. Vérifiez que vous avez activé le service dans votre compte.

Code d'état HTTP : 403

RequestAbortedException

La demande a été abandonnée avant qu'une réponse ne puisse être renvoyée. Cela se produit généralement lorsque le client ferme la connexion.

Code d'état HTTP : 400

RequestEntityTooLargeException

L'entité de demande est trop grande. Réduisez la taille du corps de la demande et réessayez.

Code d'état HTTP : 413

RequestTimeoutException

Le délai imparti pour la demande a expiré. Le serveur n'a pas reçu la demande complète dans le délai prévu. Réessayez.

Code d'état HTTP : 408

ServiceUnavailable

Le service est temporairement indisponible. Réessayez ultérieurement.

HTTP Status Code: 503

ThrottlingException

Votre taux de demandes est trop élevé. Les demandes de nouvelle tentative AWS SDKs automatique qui reçoivent cette exception. Réduisez la fréquence des requêtes.

Code d'état HTTP : 400

UnknownOperationException

L'action ou l'opération n'est pas reconnue. Vérifiez que le nom de l'action est correctement orthographié et qu'il est compatible avec la version de l'API que vous utilisez.

Code d'état HTTP :404

UnrecognizedClientException

Le certificat X.509 ou l'identifiant de clé d' AWS accès que vous avez fourni n'existe pas dans nos archives. Vérifiez que vous utilisez des informations d'identification valides et qu'elles n'ont pas expiré.

Code d’état HTTP : 403

ValidationError

L'entrée ne répond pas au format ou aux contraintes requis. Vérifiez que tous les paramètres requis sont inclus et que les valeurs sont valides.

Code d’état HTTP : 400

Paramètres communs

La liste suivante contient les paramètres que toutes les actions utilisent pour signer les demandes Signature Version 4 à l'aide d'une chaîne de requête. Tous les paramètres spécifiques d'une action particulière sont énumérées dans le sujet consacré à cette action. Pour plus d'informations sur la version 4 de Signature, consultez la section Signing AWS API du guide de l'utilisateur IAM.

X-Amz-Algorithm

Algorithme de hachage que vous avez utilisé pour créer la signature de la demande.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Valeurs valides : `AWS4-HMAC-SHA256`

Obligatoire : Conditionnelle

X-Amz-Credential

Valeur de la portée des informations d'identification, qui est une chaîne incluant votre clé d'accès, la date, la région cible, le service demandé et une chaîne de terminaison (« aws4_request »). Spécifiez la valeur au format suivant : access_key/AAAAMMJJ/région/service/aws4_request.

Pour plus d'informations, consultez la section Création d'une demande d' AWS API signée dans le guide de l'utilisateur IAM.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Date

La date utilisée pour créer la signature. Le format doit être au format de base ISO 8601 (AAAAMMJJ'T'HHMMSS'Z'). Par exemple, la date et l'heure suivantes sont une X-Amz-Date valeur valide :`20120325T120000Z`.

Condition : X-Amz-Date est facultative pour toutes les demandes ; elle peut être utilisée pour remplacer la date utilisée pour signer les demandes. Si l'en-tête Date est spécifié au format de base ISO 8601, X-Amz-Date il n'est pas obligatoire. Lorsqu'il X-Amz-Date est utilisé, il remplace toujours la valeur de l'en-tête Date. Pour plus d'informations, consultez la section Éléments d'une signature de demande d' AWS API dans le Guide de l'utilisateur IAM.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Security-Token

Le jeton de sécurité temporaire obtenu par un appel à AWS Security Token Service (AWS STS). Pour obtenir la liste des services prenant en charge les informations d'identification de sécurité temporaires d' AWS STS, consultez Services AWS qui fonctionnent avec IAM dans le Guide de l'utilisateur IAM.

Condition : Si vous utilisez des informations d'identification de sécurité temporaires provenant de AWS STS, vous devez inclure le jeton de sécurité.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Signature

Spécifie la signature codée en hexadécimal qui a été calculée à partir de la chaîne à signer et de la clé de signature dérivée.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-SignedHeaders

Spécifie tous les en-têtes HTTP qui ont été inclus dans la demande canonique. Pour plus d'informations sur la spécification d'en-têtes signés, consultez la section Création d'une demande d' AWS API signée dans le guide de l'utilisateur IAM.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

Historique du document pour Amazon Personalize

Le tableau suivant décrit les modifications importantes apportées à chaque version du guide du développeur Amazon Personalize. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
Nouvelle fonction	Amazon Personalize introduit la recette de similarité sémantique qui définit la similarité des articles en fonction de métadonné es telles que les titres et les descriptions, avec un classement optionnel par popularité ou horodatage, un support de formation progressi f et une sélection flexible des colonnes pour la formation . Pour plus d'informations, consultez la section Recette de similarité sémantique.	30 novembre 2025
Nouvelle fonction	Amazon Personalize prend désormais en charge la configuration de solutions pour attribuer une pondérati on différente aux différents types d'événements. Pour plus d'informations, consultez Optimisation d'une solution avec configuration d'événeme nts.	11 avril 2025
Nouvelle fonction	Amazon Personalize prend désormais en charge IPv6	31 janvier 2025

le trafic via des points de terminaison à double pile. Pour obtenir la liste des points de terminaison Amazon Personalize par région, consultez la section AWSRégions et points de terminaison dans le manuel Amazon Web Services General Reference.

Nouvelle fonctionnalité de documentation	Grâce aux données de consultation des pages et à une analyse approfondie, la structure du guide du développeur Amazon Personalize est mise à jour afin d'accroître la visibilité des sujets importants. La navigation a été mise à jour pour réduire la profondeur globale et les sujets connexes ont été consolidés. Par exemple, vous pouvez désormais trouver tous les conseils de préparation des données dans une rubrique Préparation des données de formation pour Amazon Personalize. Pour un sujet qui peut vous aider à naviguer dans la nouvelle structure, consultez les détails du flux de travail Amazon Personalize. Cette page associe chaque étape du flux de travail Amazon Personalize à la rubrique associée dans le guide.	3 septembre 2024

Nouvelle fonction	Vous pouvez désormais mettre à jour une solution personnalisée Amazon Personalize existante pour modifier sa configuration de formation. Vous pouvez activer ou désactiver l'entraînement automatique, et vous pouvez modifier la fréquence d'entraînement. Pour plus d'informations, voir Mettre à jour une solution pour modifier sa configuration d'entraînement automatique.	29 août 2024
Nouvelle fonctionnalité de documentation	La rubrique Personnalisation des résultats de recherche à partir de la OpenSearch rubrique du guide du développeur Amazon Personalize est réorganisée en OpenSearch fonction du service et des flux de travail open source OpenSearch . Il propose désormais de nouveaux sous-sujets : Personnalisation des résultats d'Amazon OpenSearch Service avec Amazon Personalize et Personnalisation des résultats d'Open Source Open Search avec Amazon Personalize.	23 août 2024

Nouvelle fonction	Amazon Personalize permet désormais de supprimer des utilisateurs et leurs données, y compris leurs métadonnées et leurs données d'interaction, d'un groupe de données. Pour plus d'informations, consultez la section Suppression d'utilisateurs.	2 mai 2024
Nouvelle fonction	Amazon Personalize peut désormais prendre en compte jusqu'à 5 millions d'articles grâce à une formation plus rapide, à une latence de recommandation plus faible et à des recommandations plus pertinentes. Deux nouvelles recettes personnalisées utilisent cette formation améliorée, User-Personalization-v2 et Personalized-Ranking-V2.	1 mai 2024
Nouvelle fonction	Toutes les nouvelles solutions personnalisées Amazon Personalize utilisent désormais la formation automatique. Pour plus d'informations sur la configuration de l'entraînement automatique, voir Configuration de l'entraînement automatique.	19 avril 2024

Nouvelle fonction	Amazon Personalize permet désormais de recommander la meilleure action suivante à un utilisateur en fonction de son comportement. Pour plus d'informations, consultez Next-Best-Action la recette.	26 novembre 2023
Nouvelle fonction	Amazon Personalize permet désormais d'inclure des thèmes descriptifs dans les recommandations par lots à l'aide de l'IA générative. Pour plus d'informations sur la génération de recommandations par lots avec des thèmes, consultez la section Recommandations par lots avec des thèmes. Pour plus d'informations sur l'utilisation d'Amazon Personalize avec l'IA générative, consultez Amazon Personalize et l'IA générative.	26 novembre 2023

Nouvelle fonction	Amazon Personalize permet désormais d'inclure les métadonnées d'un ensemble de données Items dans les recommandations. Pour plus d'informations sur l'activation des métadonnées pour une campagne, voir les métadonnées des éléments dans les recommandations. Pour plus d'informations sur l'activation des métadonnées pour un recommandeur, consultez la section Activation des métadonnées dans les recommandations pour un recommandeur.	26 novembre 2023
Nouvelle fonction	Amazon Personalize permet désormais de personnaliser les résultats de recherche à partir du OpenSearch Service. Pour plus d'informations, consultez la section Personnalisation des résultats de recherche à partir de OpenSearch.	16 octobre 2023

Nouvelle fonction	Amazon Personalize permet désormais d'importer jusqu'à 100 colonnes de métadonnées pour les ensembles de données Items (au lieu de 50) et 25 colonnes de métadonnées pour les ensembles de données Users (au lieu de 5). Pour plus d'informations sur les limites d'Amazon Personalize, consultez Amazon Personalize endpoints and quotas.	5 septembre 2023
Nouvelle fonction	Amazon Personalize permet désormais de personnaliser les résultats de recherche à partir de OpenSearch. Pour plus d'informations, voir Personnalisation des résultats de recherche depuis OpenSearch (autogéré).	25 juillet 2023
Nouvelle fonction	Amazon Personalize permet désormais de remplacer le schéma d'un ensemble de données par un schéma nouveau ou existant. Pour plus d'informations, consultez la section Remplacement de la structure d'un ensemble de données.	13 juillet 2023

Nouvelle fonction	Si vous utilisez les recettes User-Personalization ou Personalized-Ranking, les tâches d'inférence par lots Amazon Personalize peuvent désormais utiliser des données que vous importez de manière incrémentielle sans formation préalable . Pour plus d'informations, consultez la section Obtenir des recommandations par lots.	30 juin 2023
Nouvelle fonction	Amazon Personalize permet désormais de filtrer les articles en fonction de l'article que vous spécifiez dans votre demande de recommand ations d'articles connexes. Pour plus d'informations sur les filtres, consultez la section Recommandations de filtrage et segments d'utilisateurs.	21 juin 2023
Nouvelle fonction	Amazon Personalize permet désormais de filtrer les articles en fonction de l'article que vous spécifiez dans votre demande de recommand ations d'articles connexes. Pour plus d'informations sur les filtres, consultez la section Recommandations de filtrage et segments d'utilisateurs.	21 juin 2023

Nouvelle fonction	Amazon Personalize prend désormais en charge les connexions privées entre un cloud privé virtuel (VPC) et Amazon Personalize grâce à une interface Amazon VPC endpoint. Pour plus d'informations, consultez Amazon Personalize et interfacez les points de terminaison VPC (). AWS PrivateLink	12 juin 2023
Nouvelle fonction	Amazon Personalize prend désormais en charge la configuration des colonnes utilisées lors de la formation lorsque vous créez un recommandateur ou une solution personnalisée. Pour plus d'informations sur la configuration des colonnes lors de la création d'une recommandation, consultez la section Création de recommandations. Pour plus d'informations sur la configuration des colonnes lors de la création d'une solution, voir Configuration des colonnes utilisées lors de la formation.	30 mai 2023

Nouvelle fonctionnalité de documentation	Le guide du développeur Amazon Personalize inclut désormais des informations sur la réalisation de A/B tests à l'aide des recommandations d'Amazon Personalize. Pour plus d'informations, consultez la section Mesurer l'impact des recommandations à l'aide A/B de tests.	5 mai 2023
Nouvelle fonction	Amazon Personalize permet désormais de configurer l'influence de la popularité sur les recommandations générées par la recette Similar-Items. Pour plus d'informations, consultez la section Recette d'articles similaires.	21 avril 2023
Nouvelle fonction	Amazon Personalize prend désormais en charge l'utilisation d'Amazon SageMaker AI Data Wrangler pour importer des données provenant de plus de 40 sources dans des ensembles de données Amazon Personalize. Pour plus d'informations, consultez Importation de données à l'aide d'Amazon SageMaker AI Data Wrangler.	14 avril 2023

Nouvelle fonctionnalité de documentation	Le guide du développeur Amazon Personalize inclut désormais une nouvelle liste de contrôle de préparation. Cette liste de contrôle vous aide à vous préparer à utiliser Amazon Personalize avec vos propres données. Pour plus d'informations, consultez la liste de contrôle de préparation.	9 février 2023
Nouvelle fonction	Amazon Personalize prend désormais en charge la génération d'informations et de statistiques pour les données que vous importez dans des ensembles de données. Pour plus d'informations, consultez la section Analyse des données dans des ensembles de données.	25 janvier 2023
Nouvelle fonction	Amazon Personalize prend désormais en charge une nouvelle recette Trending-Now pour les groupes de données personnalisés. Pour plus d'informations, consultez la recette Trending-Now.	6 janvier 2023

Nouvelle fonction	Amazon Personalize prend désormais en charge l'utilisation de balises dans les politiques IAM. Pour plus d'informations, consultez la section Utilisation de balises dans les politiques IAM.	28 décembre 2022
Nouvelle fonction	Amazon Personalize permet désormais d'ajuster le nombre maximum d'interactions prises en compte par un modèle pendant le quota d'entraînement. En outre, le quota limite du nombre maximum d'utilisateurs pris en compte par un modèle pendant l'entraînement ne s'applique plus. Pour plus d'informations, consultez Amazon Personalize pour les points de terminaison et les quotas.	15 décembre 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge la création d'une attribution métrique pour mesurer l'impact commercial des recommandations. Pour plus d'informations, consultez la section Mesurer l'impact des recommandations.	17 novembre 2022

Nouvelle fonction	Les quotas Amazon Personalize relatifs au nombre total de solutions actives, de campagnes actives, de recommandations et de filtres ont désormais été augmentés. Chacun de ces quotas s'applique désormais par groupe de jeux de données plutôt que par compte. Pour plus d'informations sur les quotas, consultez Amazon Personalize endpoints and quotas.	7 septembre 2022
Nouvelle fonction	Les filtres Amazon Personalize prennent désormais en compte jusqu'à 100 interactions par utilisateur et par type d'événement. Pour plus d'informations sur les filtres, consultez la section Recommandations de filtrage et segments d'utilisateurs.	29 août 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge un nouveau cas d'utilisation Trending now pour le domaine VIDEO_ON_DEMAND. Pour plus d'informations, consultez les cas d'utilisation de VIDEO_ON_DEMAND.	17 août 2022

Nouvelle fonction	Amazon Personalize prend désormais en charge la promotion d'articles dans les recommandations grâce à un filtre de promotion distinct. Pour plus d'informations sur la promotion d'articles, voir Promouvoir des articles dans les recommandations.	12 août 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge l'utilisation d'opérateurs de comparaison dans les expressions de filtre avec des paramètres d'espace réservé. Pour plus d'informations sur les expressions de filtre, consultez la section Expressions de filtre.	12 août 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge les mises à jour groupées incrémentielles des ensembles de données. Vous pouvez désormais utiliser une tâche d'importation de jeu de données pour mettre à jour un ensemble de données sans remplacer les données existantes. Pour plus d'informations, consultez la section Mise à jour de données groupées existantes.	2 août 2022

Nouvelle fonction	Amazon Personalize peut désormais utiliser des métadonnées de texte non structurées dans différentes langues. Pour plus d'informations, consultez la section Métadonnées de texte non structuré.	6 juin 2022
Nouvelle fonction	Les recommandations d'Amazon Personalize génèrent désormais des statistiques hors ligne. Vous pouvez utiliser ces mesures pour évaluer les performances de votre recommandeur. Pour plus d'informations, consultez la section Évaluation d'un recommandeur.	24 mai 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge la possibilité d'arrêter un recommandeur et de le redémarrer ultérieurement. Ainsi, vous pouvez suspendre la facturation du recommandeur et ne le payer que lorsque vous l'utilisez. Pour plus d'informations, consultez la section Arrêt et démarrage d'un recommandeur.	20 avril 2022

Nouvelle fonction	Amazon Personalize prend désormais en charge l'utilisation de balises pour classer et gérer les ressources Amazon Personalize. Pour plus d'informations, consultez la section Marquage des ressources Amazon Personalize.	7 avril 2022
Nouvelle fonction	Amazon Personalize prend désormais en charge la spécification de ressources avecCloudFormation. Pour plus d'informations, consultez la section Spécification des ressources avec CloudFormation.	11 mars 2022
Nouvelle fonctionnalité de documentation	Le guide du développeur Amazon Personalize inclut désormais une nouvelle rubrique de résolution des problèmes qui fournit des réponses aux questions courantes et des conseils de résolution des problèmes liés aux messages d'erreur que vous pourriez rencontrer avec Amazon Personalize. Pour plus d'informations, consultez Dépannage.	15 février 2022

Nouvelle fonction	Amazon Personalize prend désormais en charge la création d'un groupe de données de domaines avec des ressources optimisées pour les cas d'utilisation pour les domaines de vidéo à la demande ou de commerce électronique. Pour plus d'informations, consultez la section Groupes de jeux de données de domaine.	29 novembre 2021
Nouvelle fonction	Amazon Personalize prend désormais en charge la création de segments d'utilisateurs à l'aide de nouvelles recettes USER_SEGMENTATION. Les recettes USER_SEGMENTATION génèrent des segments d'utilisateurs en fonction des données d'entrée des articles. Pour plus d'informations, voir recette et Item-Attribute-Affinity recette Item-Affinity.	29 novembre 2021

Nouvelle fonction	Amazon Personalize prend désormais en charge une nouvelle recette de recommandation RELATED_ITEMS, Similar-Items. Utilisez la recette Similar-Items pour générer des recommandations pour des éléments similaires en fonction à la fois des données d'interaction et des métadonnées des articles. Pour plus d'informations, consultez la section Recette d'articles similaires.	5 octobre 2021
Nouvelle fonctionnalité de documentation	Le guide du développeur Amazon Personalize inclut désormais un didacticiel de démarrage pour l'utilisation d'Amazon Personalize avec le SDK pour Java 2.x. Pour plus d'informations, consultez Getting started (SDK for Java 2.x).	25 août 2021
Nouvelle fonction	Amazon Personalize peut désormais extraire des informations pertinentes à partir de métadonnées de texte non structurées dans un ensemble de données Items. Pour plus d'informations, consultez la section Ensemble de données Items.	9 juin 2021

Nouvelle fonction	Amazon Personalize permet désormais d'arrêter de créer une version de solution (arrêter de former un modèle). Pour plus d'informations, consultez la section Arrêt de la création d'une version de solution.	20 mai 2021
Nouvelle fonctionnalité (version préliminaire)	Amazon Personalize peut désormais optimiser une solution pour un objectif en plus de maximiser la pertinence, par exemple en maximisant les revenus. Cette fonctionnalité est en version préliminaire. Pour plus d'informations, voir Optimisation d'une solution pour un objectif supplémentaire.	18 mai 2021
Nouvelle fonction	Amazon Personalize peut désormais exporter les enregistrements d'un ensemble de données Amazon Personalize vers un compartiment Amazon S3 à des fins d'analyse et de suivi. Pour plus d'informations, consultez la section Exportation d'un ensemble de données.	26 avril 2021

Nouvelle fonction	Amazon Personalize met désormais automatiquement à jour le dernier modèle (version de solution) que vous avez formé avec User-Personalization toutes les deux heures afin d'inclure de nouvelles données. Pour plus d'informations, consultez la section Recette de personnalisation utilisateur.	17 novembre 2020
Nouvelle fonction	Amazon Personalize peut désormais filtrer les recommandations en fonction de critères que vous spécifiez lorsque vous recevez des recommandations. Pour plus d'informations, consultez la section Recommandations relatives au filtrage.	10 novembre 2020
Nouvelle fonction	Amazon Personalize permet désormais d'importer progressivement des utilisateurs et des articles. Pour plus d'informations, consultez Importer des enregistrements de manière incrémentielle.	2 octobre 2020

Nouvelle fonction	Amazon Personalize prend désormais en charge une nouvelle recette de recommandation USER_PERSONALIZATION. Les fonctionnalités USER_PERSONALIZATION incluent la modélisation des données d'empreinte, l'exploration automatique des objets et la sélection automatique des articles à froid. Pour plus d'informations, consultez la section Recette de personnalisation utilisateur.	5 août 2020
Nouvelle fonction	Amazon Personalize peut désormais filtrer les recommandations en fonction des métadonnées des articles et des utilisateurs à l'aide d'expressions de filtre personnalisées. Pour plus d'informations, consultez la section Recommandations relatives au filtrage.	31 juillet 2020
Nouvelle fonction	Amazon Personalize vous permet désormais de filtrer les résultats en fonction des éléments avec lesquels un utilisateur a interagi. Pour plus d'informations, consultez la section Recommandations relatives au filtrage.	3 juin 2020

Nouvelle fonction	Amazon Personalize publie désormais les scores des articles recommandés. Les scores indiquent que le modèle Amazon Personalize est certain qu'un utilisateur choisira ensuite un article en particulier. Pour plus d'informations, consultez la section Obtenir des recommandations.	3 avril 2020
Nouvelle région	Amazon Personalize ajoute la prise en charge de la région Asie-Pacifique (Séoul). Pour obtenir la liste complète des AWS régions prises en charge par Amazon Personalize, consultez le AWStableau des AWSrégions ou les régions et points de terminaison dans le manuel Amazon Web Services General Reference.	21 janvier 2020
Nouvelle fonction	Amazon Personalize peut désormais obtenir des recommandations basées sur des métadonnées contextuelles. Pour plus d'informations, consultez la section Obtenir des recommandations.	19 décembre 2019

Nouvelles régions	Amazon Personalize ajoute la prise en charge des régions Asie-Pacifique (Mumbai), Asie-Pacifique (Sydney) et Canada (centre). Pour obtenir la liste complète des AWS régions prises en charge par Amazon Personalize, consultez le AWStableau des AWSrégions ou les régions et points de terminaison dans le manuel Amazon Web Services General Reference.	18 décembre 2019
Nouvelle fonction	Amazon Personalize prend désormais en charge les flux de travail de recommandation par lots. Pour plus d'informa tions, consultez la section Obtenir des recommandations par lots.	14 novembre 2019
Disponibilité générale d'Amazon Personalize	Amazon Personalize est désormais disponible pour une utilisation générale.	10 juin 2019
Version préliminaire d'Amazon Personalize	Il s'agit de la première version préliminaire de la documenta tion d'Amazon Personalize.	28 novembre 2018

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.