aws

Guía para desarrolladores

Amazon Personalize



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

Amazon Personalize: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon Personalize? 1
Precios de Amazon Personalize 2
Guía para usuarios nuevos 3
Descubrimiento de Amazon Personalize con Magic Movie Machine 3
Navegación por los materiales de introducción de esta guía 3
Relacionado AWS servicios y soluciones 4
Third-party servicios 4
Más información 5
Trabajando con AWS SDK 5
Amazon Personalize y la IA generativa 7
Recomendaciones con temas del Generador de contenidos 7
Metadatos de las recomendaciones 8
Código LangChain preconfigurado para la personalización 9
Funcionamiento 11
Flujo de trabajo de Amazon Personalize 11
Términos de Amazon Personalize 13
Importación y administración de datos 13
Formación 16
Implementación y recomendaciones del modelo 19
Datos de Amazon Personalize 20
Datos de interacciones 21
Datos de elementos 21
Datos de usuario 21
Datos de acciones 22
Datos de interacciones de acción 22
Configuración de Amazon Personalize 23
Inscríbase para obtener una Cuenta de AWS 23
Regiones y puntos de conexión 23
Configuración de permisos 23
Concesión de permiso a Amazon Personalize para que acceda a sus recursos 24
Concesión de permiso a los usuarios para acceder a Amazon Personalize 28
Concesión de acceso a Amazon Personalize para los recursos de Amazon S3 30
Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave 37
Configuración del AWS CLI 39

Configurando el AWS SDK 40
Tutoriales de introducción 42
Requisitos previos para los ejercicios de introducción 43
Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio) 43
Creación de los datos de entrenamiento (grupo de conjuntos de datos personalizado) 45
Introducción a un grupo de conjuntos de datos de dominio 45
Introducción a un grupo de conjuntos de datos de dominio (consola) 46
Introducción a un grupo de conjuntos de datos de dominio (SDK para Java 2.x) 53
Introducción a un grupo de conjuntos de datos de dominio (SDK para Python (Boto3)) 63
Introducción a un grupo de conjuntos de datos de dominio (SDK para JavaScript v3) 69
Introducción a un grupo de conjuntos de datos personalizados 77
Introducción (consola) 78
Introducción (AWS CLI) 89
Introducción (SDK para Python [Boto3]) 100
Introducción (SDK para Java 2.x) 106
Correlación del caso de uso con los recursos de Amazon Personalize 119
Características del caso de uso y de la receta 120
Personalización en tiempo real 120
Exploration (Exploración) 121
Actualizaciones automáticas 123
Elección de un caso de uso 124
Casos de uso de VIDEO_ON_DEMAND 125
Casos de uso de ECOMMERCE 130
Elección de una receta 134
Tipos de recetas por caso de uso de Amazon Personalize 135
Recetas de Amazon Personalize 136
Visualización de las recetas de Amazon Personalize disponibles 139
User-Personalization-v2 139
User-Personalization 143
Trending-Now 159
Popularity-Count 163
Personalized-Ranking-v2 164
Personalized-Ranking 168
Similitud semántica 175
Similar-Items 179
SIMS 183

Next-Best-Action 190
Item-Affinity 195
Item-Attribute-Affinity 197
Recetas heredadas 199
Preparación de los datos de entrenamiento 221
Directrices de formato de datos masivos para todos los tipos de datos 222
Datos de interacciones de elementos 224
Requisitos de datos de interacciones de elemento 225
Datos de marca temporal 226
Datos de valor de evento y tipo de evento 226
Metadatos contextuales 228
Datos de impresiones 229
Ejemplo de datos de interacciones 233
Metadatos de elemento 234
Requisitos de datos de elemento 235
Datos de la marca de tiempo de creación 238
Metadatos categóricos 238
Metadatos de texto no estructurado 239
Datos numéricos 240
Datos de cadena no categóricos 241
Ejemplo de metadatos de elementos 241
Metadatos de usuario 243
Requisitos de datos de usuario 244
Metadatos categóricos 244
Datos de cadena no categóricos 245
Ejemplo de metadatos de usuarios 245
Metadatos de acción 246
Requisitos de datos de acciones 247
Datos de la marca de tiempo de caducidad de la acción 247
Datos de frecuencia de repetición 248
Value data 248
Datos de la marca de tiempo de creación 249
Metadatos categóricos 249
Datos de cadena no categóricos 249
Ejemplo de metadatos de acción 250
Datos de interacciones de acción 251

Requisitos de datos de interacciones de acción ... 252
Datos de tipo de evento ... 253
Ejemplo de datos de interacciones de acción ... 253
Creación de archivos JSON de esquema ... 255
Requisitos de formato de esquema ... 255
Tipos de datos de esquema ... 256
Conjuntos de datos y esquemas VIDEO_ON_DEMAND ... 257
Requisitos de esquema y conjunto de datos de dominio VIDEO_ON_DEMAND ... 258
Requisitos del conjunto de datos de interacciones de elementos ... 259
Requisitos de conjuntos de datos de usuarios ... 261
Requisitos de conjuntos de datos de elementos ... 263
Conjuntos de datos y esquemas de ECOMMERCE ... 266
Requisitos de esquema y conjunto de datos del dominio de ECOMMERCE ... 267
Requisitos del conjunto de datos de interacciones de elementos ... 268
Requisitos de conjuntos de datos de usuarios ... 270
Requisitos de conjuntos de datos de elementos ... 272
Esquemas y conjuntos de datos personalizados ... 274
Requisitos de esquemas y conjuntos de datos personalizados ... 275
Requisitos del esquema del conjunto de datos de interacciones de elementos ... 278
Requisitos del esquema del conjunto de datos de usuarios ... 281
Requisitos del esquema del conjunto de datos de elementos ... 283
Requisitos del esquema del conjunto de datos de acciones ... 285
Requisitos del esquema del conjunto de datos de interacciones de acción ... 287
Lista de verificación de preparación ... 290
¿Ha adaptado sus casos de uso a los recursos de Amazon Personalize? ... 290
¿Tiene suficientes datos de interacciones de elementos? ... 291
¿Cuenta con una arquitectura de streaming de eventos en tiempo real? ... 291
¿Están sus datos optimizados para Amazon Personalize? ... 292
¿Recopila datos opcionales que puedan mejorar las recomendaciones? ... 292
¿Tiene un plan para poner a prueba sus recomendaciones? ... 293
¿Tiene objetivos empresariales adicionales? ... 293
Creación de un grupo de conjuntos de datos ... 294
Creación de un grupo de conjuntos de datos (consola) ... 295
Creación de un grupo de conjuntos de datos (AWS CLI) ... 296
Creación de un grupo de conjuntos de datos (SDK de AWS) ... 297
Creación de un esquema y un conjunto de datos ... 300

Creación de un conjunto de datos y un esquema (consola) ... 300
Creación de un conjunto de datos y un esquema (AWS CLI) ... 302
Creación de un conjunto de datos y un esquema (AWS SDKs) ... 303
Importación de datos de entrenamiento ... 309
Importación de datos masivos ... 310
Modos de importación ... 311
Creación de un trabajo de exportación de conjuntos de datos (consola) ... 312
Creación de un trabajo de importación de conjuntos de datos (AWS CLI) ... 314
Creación de un trabajo de importación de conjuntos de datos (AWS SDKs) ... 317
Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler ... 321
Información adicional ... 323
Configuración de permisos ... 323
Lanzamiento de Data Wrangler desde Amazon Personalize ... 325
Importación de datos en Data Wrangler ... 326
Transformación de datos ... 328
Generación de visualizaciones y análisis de datos ... 330
Procesamiento de los datos y su importación en Amazon Personalize ... 335
Importación de registros de individuales ... 337
Importación de interacciones de forma individual ... 338
Importación de usuarios de forma individual ... 340
Importación de elementos de forma individual ... 345
Importación de acciones de forma individual ... 350
Análisis de los datos de entrenamiento ... 354
Permisos necesarios para analizar datos ... 355
Información sobre los datos ... 355
Visualización de estadísticas e información del conjunto de datos ... 359
Recomendadores de dominios ... 361
Estados de los generadores de recomendaciones ... 362
Solicitudes de recomendación mínimas por segundo y escalado automático ... 363
Creación de generadores de recomendaciones de dominios ... 363
Creación de generadores de recomendaciones (consola) ... 364
Creación de un recomendador (AWS CLI) ... 367
Creación de un recomendador (AWS SDKs) ... 367
Habilitación de los metadatos en las recomendaciones ... 370
Configuración de las columnas que se utilizan en el entrenamiento ... 371
Configuración de la exploración de un recomendador de dominios ... 374

Evaluación de un recomendador de dominios 378
Recuperación de métricas 379
Definiciones de métricas 381
Ejemplo 383
Recursos adicionales 384
Actualización de un recomendador 384
Actualización de un recomendador (consola de Amazon Personalize) 385
Actualización de un recomendador (AWS CLI) 386
Actualización de un recomendador (SDK de AWS) 386
Detención de un recomendador 388
Detención e inicio de un recomendador (consola) 389
Detención de un recomendador (AWS CLI) 389
Detención de un recomendador (SDK de AWS) 389
Recursos personalizados 392
Configuración de una solución 393
Creación de una solución 394
Configuración del entrenamiento automático 403
Configuración de las columnas que se utilizan en el entrenamiento 408
Optimización de una solución para un objetivo adicional 410
Optimización de una solución con la configuración de eventos 417
Hiperparámetros y HPO 423
Elección de los datos de interacciones de elementos utilizados para el entrenamiento 427
Clonación de una solución (consola) 429
Actualización de una solución 430
Actualización de una solución (consola) 431
Actualización de una solución (AWS CLI) 431
Actualizar una solución (AWS SDK) 432
Creación manual de una versión de solución 432
Creación de una versión de la solución (consola) 433
Creación de una versión de solución (AWS CLI) 434
Creación de una versión de la solución AWS (SDK de) 436
Detención de la creación de una versión de solución 440
Detención de la creación de una versión de solución (consola) 440
Detención de la creación de una versión de solución (AWS CLI) 441
Detención de la creación de una versión de solución (AWS SDKs) 441
Evaluación de una versión de solución 443

Recuperación de métricas de la versión de la solución 445
Definiciones de métricas 448
Ejemplo 454
Recursos adicionales 455
Creación de una campaña 455
Actualizaciones de campaña automáticas 456
TPS mínimo aprovisionado 457
Metadatos de elemento en las recomendaciones 458
Crear una campaña (consola) 459
Creación de una campaña (AWS CLI) 461
Creación de una campaña (SDK de AWS) 463
Actualización de una campaña 465
Actualización de una campaña (consola) 466
Actualizar una campaña (AWS CLI) 467
Actualizar una campaña (AWS SDK) 467
Obtención de recomendaciones 470
Puntuaciones de recomendaciones 470
Recomendaciones de elementos en tiempo real 471
Cómo funciona la puntuación de las recomendaciones (recursos personalizados) 472
Motivos de recomendación con User-Personalization-v2 474
Obtención de recomendaciones de elementos en tiempo real 474
Obtención de metadatos de elemento con recomendaciones 479
Promoción de elementos 480
Recomendaciones de acción en tiempo real 491
Cómo funciona la puntuación de recomendaciones de acciones 492
Obtención de recomendaciones de acciones (consola) 492
Obtención de recomendaciones de acciones (AWS CLI) 493
Obtención de recomendaciones de acciones (SDK de AWS) 493
Obtención de una clasificación personalizada (recursos personalizados) 494
Cómo funciona la puntuación de clasificación personalizada 495
Obtención de una clasificación personalizada (consola) 495
Obtención de una clasificación personalizada (AWS CLI) 496
Obtención de una clasificación personalizada (SDK de AWS) 497
Cuaderno de ejemplo 501
Usar metadatos contextuales 501
Obtener recomendaciones mediante metadatos contextuales (SDK de AWS Python) 502

Obtención de recomendaciones de elemento por lotes ... 503
Flujo de trabajo por lotes ... 504
Directrices y requisitos ... 504
Puntuación de flujos de trabajo por lotes ... 505
Recomendaciones por lotes con temas ... 505
Preparación de los datos de entrada para las recomendaciones por lotes ... 508
Creación de un trabajo de inferencia por lotes ... 513
Ejemplos de salida de trabajo de inferencia por lotes ... 523
Obtención de segmentos de usuarios por lotes ... 525
Directrices y requisitos para obtener segmentos de usuarios ... 526
Preparación de los datos de entrada para los segmentos de usuarios ... 527
Creación de un trabajo de segmento por lotes ... 530
Ejemplos de formato de resultados de trabajos de segmento por lotes ... 537
Filtrado de resultados ... 539
Expresiones de filtro ... 540
Directrices y requisitos ... 542
Estructura y elementos de las expresiones de filtro ... 543
Ejemplos de expresiones de filtro ... 545
Filtrado de recomendaciones en tiempo real ... 551
Filtrado de recomendaciones en tiempo real (consola) ... 552
Filtrar recomendaciones en tiempo real (AWS CLI) ... 558
Filtrar recomendaciones en tiempo real (AWS SDK) ... 560
Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados) ... 566
Proporcionar valores de filtro en su JSON de entrada ... 566
Filtrado de flujos de trabajo por lotes (consola) ... 568
Filtrar flujos de trabajo por lotes (AWS SDK) ... 568
Registro de eventos ... 569
Influencia de los eventos en tiempo real en las recomendaciones ... 570
Registro de eventos de interacción de elementos ... 570
Requisitos para registrar eventos de interacción de elementos y entrenar un modelo ... 571
Creación de un rastreador de eventos de interacción de elementos ... 572
Registro de un solo evento de interacción de elementos ... 576
Registro de varios eventos de interacción de elementos con datos de valores de eventos ... 579
Registro de eventos de interacción de elemento con datos de impresiones ... 582
Informes de atribución y métricas de eventos ... 584

Registro de eventos de interacción de acciones ... 586
Requisitos para registrar eventos de interacción de acciones ... 587
ID del rastreador de eventos de interacción de acciones ... 587
Registro de un solo evento de interacción de acciones ... 588
Registro de varios eventos de interacción de acciones ... 589
Registro de eventos para usuarios anónimos ... 590
Creación de un historial de eventos continuo para usuarios anónimos ... 591
Servicios de seguimiento de eventos de terceros ... 592
Implementaciones de ejemplo ... 592
Mantenimiento de la relevancia de las recomendaciones ... 594
Mantener actualizados los conjuntos de datos ... 594
Mantenimiento de los generadores de recomendaciones de dominios ... 595
Mantenimiento de soluciones personalizadas ... 595
Actualización de datos en conjuntos de datos después del entrenamiento ... 597
Influencia de los nuevos datos en las recomendaciones en tiempo real ... 598
Nuevas interacciones ... 598
Elementos nuevos ... 599
Nuevos usuarios ... 600
Nuevas acciones ... 601
Cómo influyen los datos nuevos en las recomendaciones por lotes (recursos personalizados) ... 601
Nuevas interacciones ... 602
Nuevos usuarios ... 602
Elementos nuevos ... 603
Sustitución del esquema de un conjunto de datos ... 604
Directrices y requisitos ... 604
Sustitución del esquema de un conjunto de datos (consola) ... 606
Sustituir el esquema de un conjunto de datos (AWS CLI) ... 606
Reemplazar el esquema de un conjunto de datos (AWS SDK) ... 607
Exportar datos ... 609
Requisitos de permisos para trabajos de exportación de conjuntos de datos ... 610
Política de roles de servicio para exportar un conjunto de datos ... 610
Política de buckets de Amazon S3 para exportar un conjunto de datos ... 611
Creación de un trabajo de exportación de conjuntos de datos en Amazon Personalize ... 612
Creación de un trabajo de exportación de conjunto de datos (consola) ... 612
Crear un trabajo de exportación de conjuntos de datos (AWS CLI) ... 613

Crear un trabajo de exportación de conjuntos de datos (AWS SDK) 615
Eliminación de recursos 619
Directrices para eliminar recursos 619
Orden recomendado para la eliminación de recursos 620
Eliminación de usuarios 621
Directrices y requisitos 622
Preparación de una lista de usuarios para eliminarlos 623
Creación de un trabajo de eliminación de datos 624
Eliminación de un conjunto de datos 628
Eliminación de un conjunto de datos (consola) 629
Eliminación de un conjunto de datos (AWS CLI) 629
Eliminación de un conjunto de datos (AWS SDKs) 629
Medición del impacto de las recomendaciones 631
Medición del impacto de las recomendaciones con una atribución de métricas 632
Directrices y requisitos 633
Creación de una atribución de métricas 637
Actualización de una atribución de métricas 644
Eliminación de una atribución de métricas 650
Visualización de gráficos de datos métricos en CloudWatch 653
Publicación de informes de atribución de métricas en Amazon S3 654
Medir el impacto de las recomendaciones mediante A/B pruebas 659
Prácticas recomendadas para pruebas A/B 659
Personalización de los resultados de búsqueda desde OpenSearch 661
Ejemplo de casos de uso 662
Funcionamiento del complemento Search Ranking de Amazon Personalize 662
Información adicional 663
Requisitos del complemento 664
Personalización de los resultados de Amazon Service OpenSearch 665
Configuración de permisos 666
Instalación del complemento 672
Creación de una canalización 673
Aplicación del complemento 674
Comparación de resultados 676
Supervisión del complemento 678
Personalización de los resultados de Open Search de código abierto 679
Configuración de permisos 681

Instalación manual del complemento en un clúster de OpenSearch existente 683
Creación de un clúster nuevo e instalación del complemento con un script 683
Creación de una canalización 684
Aplicación del complemento 686
Comparación de resultados 687
Supervisión del complemento 688
Campos de clasificación de búsqueda personalizados 688
Ejemplo de métricas de canalización 690
Etiquetado de recursos 692
Directrices y requisitos 693
Información adicional 693
Adición de etiquetas a recursos de Amazon Personalize 694
Adición de etiquetas (consola) 694
Adición de etiquetas (AWS CLI) 695
Adición de etiquetas (AWS SDKs) 696
Eliminación de etiquetas de los recursos de Amazon Personalize 699
Eliminación de etiquetas (consola) 699
Eliminación de etiquetas (AWS CLI) 700
Eliminación de etiquetas (AWS SDKs) 700
Uso de etiquetas en políticas de IAM 701
Preguntas frecuentes 703
Importación y administración de datos 703
Creación de una solución y una versión de la solución 704
Implementación de modelos (campañas personalizadas) 705
Recomendaciones 706
Filtrado de recomendaciones 706
Mensajes de error comunes 708
Importación y administración de datos 708
Creación de una solución y una versión de la solución (recursos personalizados) 709
Implementación de modelos (campañas personalizadas) 710
Recomendadores (grupos de conjuntos de datos de dominios) 710
Recomendaciones 710
Filtrado de recomendaciones 711
Especificar recursos con AWS CloudFormation 712
Amazon Personalize y CloudFormation plantillas 712
CloudFormation Plantillas de ejemplo para los recursos de Amazon Personalize 712

CreateDatasetGroup ... 713
CreateDataset ... 713
CreateSchema ... 715
CreateSolution ... 715
Obtenga más información sobre CloudFormation ... 716
Ejemplos de código ... 717
Amazon Personalize ... 718
Conceptos básicos ... 719
Eventos de Amazon Personalize ... 772
Conceptos básicos ... 772
Versión ejecutable de Amazon Personalize ... 780
Conceptos básicos ... 781
Seguridad ... 790
Protección de datos ... 791
Cifrado de datos en Amazon Personalize ... 792
Gestión de identidad y acceso ... 793
Público ... 793
Autenticación con identidades ... 794
Administración del acceso con políticas ... 795
Cómo funciona Amazon Personalize con IAM ... 797
Cross-service prevención confusa de diputados ... 803
Identity-based ejemplos de políticas ... 804
Resolución de problemas ... 810
Monitorear con CloudWatch ... 812
Uso de CloudWatch métricas para Amazon Personalize ... 812
Acceso a métricas de Amazon Personalize ... 813
Crear una alarma de ... 814
Aplicación de monitorización ... 816
CloudWatch métricas de Amazon Personalize ... 816
Registro de llamadas a la API Amazon Personalize con AWS CloudTrail ... 821
Amazon Personalize la información en CloudTrail ... 822
Ejemplo: Entradas del archivo de registro de Amazon Personalize ... 823
Validación de conformidad ... 824
Resiliencia ... 824
Seguridad de la infraestructura ... 825
Puntos de conexión de VPC (AWS PrivateLink) ... 825

Creación de un punto de conexión de la VPC de interfaz para Amazon Personalize 826
Creación de una política de puntos de conexión de VPC para Amazon Personalize 827
Cuotas y puntos de conexión 829
Puntos de conexión y regiones de Amazon Personalize 829
Conformidad 829
Cuotas de servicio 829
Solicitud de aumento de cuota 839
Referencia de la API 841
Acciones 841
Amazon Personalize 844
Eventos de Amazon Personalize 1104
Versión ejecutable de Amazon Personalize 1120
Data Types 1138
Amazon Personalize 1141
Eventos de Amazon Personalize 1310
Versión ejecutable de Amazon Personalize 1326
Tipos de errores comunes 1332
Parámetros comunes 1335
Historial de revisión 1338
..... mccclxi

¿Qué es Amazon Personalize?

Amazon Personalize es un servicio de machine learning totalmente administrado que utiliza sus datos para generar recomendaciones de elementos para sus usuarios. También puede generar segmentos de usuarios en función de la afinidad de los usuarios con determinados elementos o con sus metadatos.

Entre los casos de uso comunes se incluyen los siguientes:

- Personalización de una aplicación de streaming de vídeo: puede utilizar los recursos de Amazon Personalize preconfigurados o personalizables para añadir varios tipos de recomendaciones de vídeo personalizadas a su aplicación de streaming. Por ejemplo, las recomendaciones de vídeo Mejores opciones, Más como X y Más popular.
- Añadir recomendaciones de productos a una aplicación de comercio electrónico: puede utilizar los recursos de Amazon Personalize preconfigurados o personalizables para añadir varios tipos de recomendaciones de productos personalizadas a su aplicación de venta minorista. Por ejemplo, las recomendaciones de productos “Recomendaciones”, “Con frecuencia se compran juntos” y “Los clientes que vieron X también vieron”.
- Adición de recomendaciones de siguiente mejor acción en tiempo real a su aplicación: puede utilizar los recursos de Amazon Personalize personalizables para recomendar las acciones que es más probable que realicen los usuarios en función de su comportamiento. Por ejemplo, puede agregar recomendaciones en tiempo real para inscribirse en el programa de fidelización, descargar la aplicación móvil o suscribirse a correos electrónicos promocionales.
- Creación de correos electrónicos personalizados: puede utilizar los recursos personalizables de Amazon Personalize para generar recomendaciones por lotes para todos los usuarios de una lista de correo electrónico. Luego, puede usar un servicio AWS o un servicio de terceros para enviar a los usuarios correos electrónicos personalizados recomendando elementos de su catálogo.
- Creación de una campaña de marketing dirigida: puede usar Amazon Personalize para generar segmentos de usuarios que probablemente interactúen con los elementos de su catálogo. Luego, puede usar un servicio AWS o un servicio de terceros para crear una campaña de marketing dirigida que promueva diferentes elementos en diferentes segmentos de usuarios.
- Personalización de los resultados de búsqueda: puede utilizar los recursos personalizables de Amazon Personalize para personalizar los resultados de búsqueda para sus usuarios. Por ejemplo, Amazon Personalize puede volver a clasificar los resultados de búsqueda que generes con OpenSearchellos.

En la mayoría de los casos de uso, Amazon Personalize genera recomendaciones basadas principalmente en los datos de interacciones de elementos. Los datos de interacciones de elementos provienen de los usuarios que interactúan con los elementos de su catálogo. Por ejemplo, los usuarios hacen clic en diferentes elementos. Los datos de interacciones de elementos pueden proceder tanto de sus registros históricos de interacciones masivas en un archivo CSV como de eventos en tiempo real de los usuarios mientras interactúan con su catálogo. En algunos casos, Amazon Personalize también utiliza datos de los elementos y los usuarios, como el género, el precio o el sexo. Además, en los escenarios de siguiente mejor acción, usa datos de acciones y de interacciones de acción.

Al importar datos masivos, puede usar Amazon SageMaker AI Data Wrangler para importar datos de más de 40 fuentes y prepararlos para Amazon Personalize. Para obtener más información, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.

Amazon Personalize incluye operaciones de API para la personalización en tiempo real y operaciones por lotes para recomendaciones masivas y segmentos de usuarios. Puede empezar rápidamente con recomendadores optimizados para casos de uso de su dominio empresarial o puede crear sus propios recursos personalizados configurables.

Temas

- Precios de Amazon Personalize
- Guía para usuarios nuevos de Amazon Personalize
- Relacionado AWS servicios y soluciones
- Third-party servicios
- Más información
- Uso de Amazon Personalize con un AWS SDK

Precios de Amazon Personalize

Con Amazon Personalize, no se requieren pagos mínimos ni compromisos iniciales. La capa AWS gratuita ofrece una cuota mensual de hasta 20 GB de procesamiento de datos por AWS región disponible, hasta 100 horas de formación por AWS región elegible y hasta 180 000 solicitudes de recomendación. El nivel gratuito es válido durante los dos primeros meses de uso.

Para obtener una lista completa de los cargos y precios, consulte Precios de Amazon Personalize.

Guía para usuarios nuevos de Amazon Personalize

Si es la primera vez que usa Amazon Personalize, los siguientes recursos le pueden ayudar a comenzar.

Temas

- Descubrimiento de Amazon Personalize con Magic Movie Machine
- Navegación por los materiales de introducción de esta guía

Descubrimiento de Amazon Personalize con Magic Movie Machine

Magic Movie Machine es una experiencia de aprendizaje interactiva. Le ayuda a descubrir las características de Amazon Personalize y a obtener más información sobre cómo generar recomendaciones. Para ver una breve introducción, consulte el siguiente vídeo. Pruebe entonces Magic Movie Machine.

Introducción a Amazon Personalize

Navegación por los materiales de introducción de esta guía

Consulte las siguientes secciones para obtener una introducción a Amazon Personalize. Para obtener una lista de verificación que proporciona listas de las características, los requisitos y la orientación sobre datos de Amazon Personalize, consulte Lista de verificación de preparación.

1. Cómo funciona Amazon Personalize: en esta sección se presenta el flujo de trabajo de Amazon Personalize y se explican los pasos para crear experiencias personalizadas para los usuarios. En esta sección también se incluyen los términos más comunes de Amazon Personalize y sus definiciones. Comience con esta sección para asegurarse de que entiende bien los flujos de trabajo y los términos de Amazon Personalize antes de empezar a recibir recomendaciones.
2. Configuración de Amazon Personalize— En esta sección Cuenta de AWS, configuras los permisos necesarios para usar Amazon Personalize y configuras los SDK AWS CLI y los AWS SDK para usar y gestionar Amazon Personalize.
3. Tutoriales de introducción: en esta sección, comenzará a usar Amazon Personalize con un conjunto de datos de películas sencillo. Complete estos tutoriales para obtener experiencia práctica con Amazon Personalize. Puede elegir empezar con un grupo de conjuntos de datos de dominio o con un grupo de conjuntos de datos personalizado:

- Para empezar a crear un grupo de conjuntos de datos de dominio, complete Requisitos previos para los ejercicios de introducción y, a continuación, inicie los tutoriales que aparecen en Introducción a un grupo de conjuntos de datos de dominio.
- Para empezar a crear un grupo de conjuntos de datos personalizado, complete Requisitos previos para los ejercicios de introducción y, a continuación, inicie los tutoriales que aparecen en Introducción a un grupo de conjuntos de datos de dominio.

Después de completar el ejercicio de introducción, puede empezar a completar el flujo de trabajo de Amazon Personalize. Para ver los pasos y los enlaces a la documentación, consulte Flujo de trabajo de Amazon Personalize.

Relacionado AWS servicios y soluciones

Amazon Personalize se integra perfectamente con otros AWS servicios y soluciones. Por ejemplo, puede hacer lo siguiente:

- Utilice Amazon SageMaker AI Data Wrangler (Data Wrangler) para importar datos de más de 40 fuentes a un conjunto de datos de Amazon Personalize. Data Wrangler es una función de Amazon SageMaker AI Studio que proporciona una solución integral para importar, preparar, transformar y analizar datos. Para obtener más información, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.
- Se usa AWS Amplify para registrar eventos de interacción entre elementos. Amplify incluye una JavaScript biblioteca para grabar eventos desde aplicaciones de cliente web. Y también incluye una biblioteca para registrar los eventos en el código del servidor. Para obtener más información, consulte Documentación de Amplify.
- Automatice y programe las tareas de Amazon Personalize con Mantenimiento de experiencias personalizadas con machine learning. Esta implementación de soluciones de AWS automatiza el flujo de trabajo de Amazon Personalize, incluida la importación de datos, el entrenamiento sobre las versiones de solución y los flujos de trabajo por lotes.

Third-party servicios

Amazon Personalize funciona bien con varios servicios de terceros.

- Amplitude: puede utilizar Amplitude para realizar un seguimiento de las acciones de los usuarios y así comprender su comportamiento. Para obtener información sobre el uso de Amplitude y

Amazon Personalize, consulte la siguiente entrada del blog de la red de socios de AWS : Medición de la eficacia de la personalización con Amplitude y Amazon Personalize.

- Braze: puede usar Braze para enviar a los usuarios correos electrónicos personalizados recomendando elementos de su catálogo. Braze es una plataforma de mensajería líder en el mercado (correo electrónico, inserción, SMS). Para ver un taller que muestra cómo integrar Amazon Personalize y Braze, consulte el taller de Amazon Personalize.
- Optimizely: puedes usar Optimizely para realizar pruebas con las recomendaciones de A/B Amazon Personalize.
- Segment: puede usar Segment para enviar sus datos a Amazon Personalize. Para obtener más información sobre la integración de Segment con Amazon Personalize, consulte Destino de Amazon Personalize.

Para obtener una lista completa de socios, consulte Socios de Amazon Personalize.

Más información

En los siguientes recursos se proporciona información adicional acerca de Amazon Personalize:

- Para obtener una referencia rápida que le ayude a determinar si Amazon Personalize se ajusta a su caso de uso, consulta la hoja de referencia de Amazon Personalize en el repositorio de muestras de Amazon Personalize.
- Para ver una serie de vídeos sobre cómo utilizar Amazon Personalize, consulte la serie de vídeos Amazon Personalize Deep Dive que se encuentra en YouTube.
- Para ver tutoriales detallados y ejemplos de código, consulte el repositorio GitHub amazon-personalize-samples.

Uso de Amazon Personalize con un AWS SDK

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código

Documentación de SDK	Ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python v4	AWS Ejemplos de código de SDK para Python v4
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Amazon Personalize y la IA generativa

Amazon Personalize funciona bien con la inteligencia artificial generativa (IA generativa). Con la ayuda de la IA generativa, el Generador de contenidos de Amazon Personalize puede agregar temas interesantes a las recomendaciones por lotes de elementos relacionados. Generador de contenidos es una capacidad de IA generativa administrada por Amazon Personalize.

También puede utilizar las recomendaciones de Amazon Personalize para integrar este servicio con su flujo de trabajo de IA generativa y mejorar la experiencia de los usuarios. Por ejemplo, puede agregar recomendaciones a los mensajes de IA generativa para crear contenido de marketing adaptado a los intereses de cada uno de los usuarios. También puede generar resúmenes concisos para el contenido recomendado, o bien recomendar productos o contenido a través de chatbots.

En el siguiente video, se muestra cómo mejorar las recomendaciones con Amazon Personalize y la IA generativa.

Mejora de las recomendaciones con Amazon Personalize y la IA generativa

Las siguientes características de Amazon Personalize utilizan la IA generativa o pueden ayudarle a crear soluciones de IA generativa que creen contenido personalizado. Para ver cuadernos de Jupyter de muestra que presentan cómo usar Amazon Personalize con la IA generativa, consulte Generative AI with Amazon Personalize en el repositorio de muestras de Amazon Personalize.

Temas

- Recomendaciones con temas del Generador de contenidos
- Metadatos de las recomendaciones
- Código LangChain preconfigurado para la personalización

Recomendaciones con temas del Generador de contenidos

El Generador de contenidos de Amazon Personalize puede agregar temas descriptivos a las recomendaciones por lotes. Generador de contenidos es una capacidad de IA generativa administrada por Amazon Personalize.

Cuando recibe recomendaciones por lotes con temas, el Generador de contenidos de Amazon Personalize agrega un tema descriptivo para cada conjunto de elementos similares. Por ejemplo, si recibe recomendaciones de elementos similares a una cierta comida para desayuno, Amazon Personalize puede generar un tema como Levántese con energía o Fundamental por la

mañana. Puede usar el tema para reemplazar un título genérico de carrusel, como Con frecuencia se compran juntos. Otra opción es incorporar el tema a un correo electrónico promocional o una campaña de marketing para crear nuevas opciones de menú.

Para generar temas, debe importar los datos en las interacciones y los conjuntos de datos de elementos, crear una solución personalizada con la receta Similar-Items y generar recomendaciones por lotes. Los datos del elemento deben incluir una descripción de este e información del título. Las descripciones y los títulos de elementos detallados ayudan al Generador de contenidos a crear temas más precisos e interesantes.

- Para obtener más información acerca del flujo de trabajo de Amazon Personalize, consulte Detalles del flujo de trabajo de Amazon Personalize.
- Para obtener información sobre las recomendaciones por lotes, consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.
- Para obtener información sobre la generación de recomendaciones de elementos con temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

Metadatos de las recomendaciones

Cuando reciba recomendaciones, puede hacer que Amazon Personalize devuelva metadatos sobre cada elemento recomendado de su conjunto de datos de elementos. Puede agregar estos metadatos, junto con las recomendaciones de Amazon Personalize, a los mensajes de la IA generativa a fin de generar contenido más atractivo.

Por ejemplo, puede utilizar la IA generativa para crear correos de marketing. Puede utilizar las recomendaciones de Amazon Personalize y sus metadatos (por ejemplo, los géneros cinematográficos) como parte de la ingeniería de peticiones para la IA generativa. Con las peticiones personalizadas, puede usar la IA generativa para crear correos electrónicos de marketing atractivos y adaptados a los intereses de cada uno de sus clientes.

Para obtener los metadatos de las recomendaciones, primero debe completar el flujo de trabajo de Amazon Personalize para importar datos y crear recursos de dominio o personalizados. Al crear un recomendador o una campaña de Amazon Personalize, habilite la opción para incluir metadatos en las recomendaciones. Cuando reciba recomendaciones, puede especificar qué columnas de datos de los elementos quiere incluir.

- Para obtener más información acerca del flujo de trabajo de Amazon Personalize, consulte Detalles del flujo de trabajo de Amazon Personalize.

- Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones (recursos de dominio).
- Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Habilitación de los metadatos en las recomendaciones (recursos personalizados).
- Para obtener más información sobre cómo utilizar Amazon Personalize con la IA generativa para crear campañas de marketing, consulte Mejore sus soluciones de marketing con Amazon Personalize y la IA generativa.

Código LangChain preconfigurado para la personalización

LangChain es un marco para desarrollar aplicaciones basadas en modelos de lenguaje. Incluye código creado para Amazon Personalize. Puede usar este código para integrar las recomendaciones de Amazon Personalize en su solución de IA generativa.

Por ejemplo, puede utilizar el código siguiente para agregar a su cadena las recomendaciones de Amazon Personalize para un usuario.

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

recommender_arn="RECOMMENDER ARN"

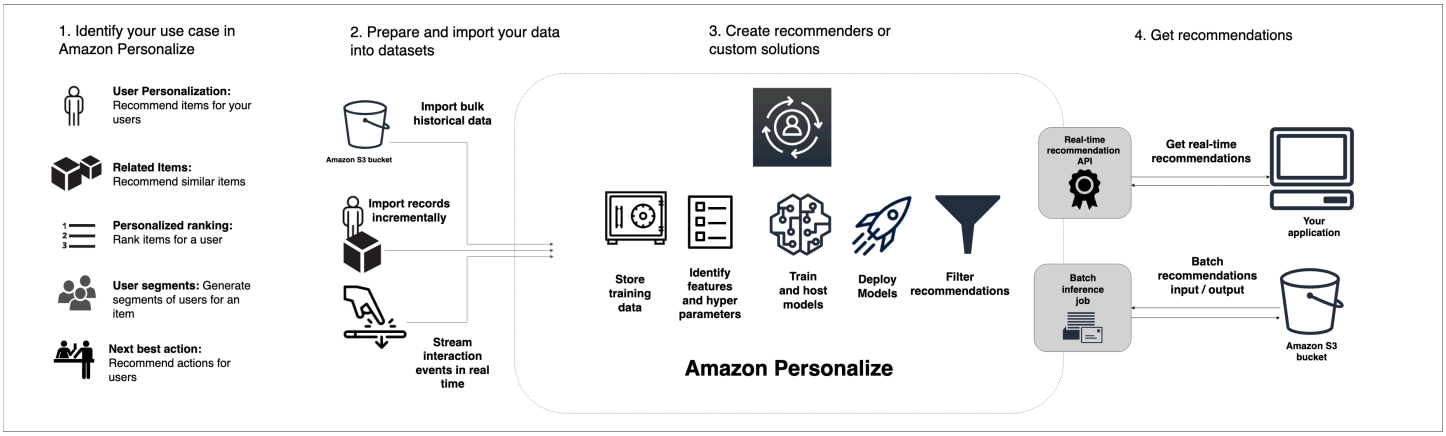
bedrock_llm = Bedrock(model_id="anthropic.claude-v2", region_name="us-west-2")
client=AmazonPersonalize(credentials_profile_name="default",region_name="us-
west-2",recommender_arn=recommender_arn)
# Create personalize chain
# Use return_direct=True if you do not want summary
chain = AmazonPersonalizeChain.from_llm(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

- Para obtener información sobre cómo empezar a utilizar LangChain, consulte Introducción en la documentación de LangChain.

- Para obtener información sobre el uso de código LangChain para Amazon Personalize, incluidos ejemplos de código más avanzados, consulte las extensiones LangChain de Amazon Personalize en el repositorio de ejemplos de AWS.

Cómo funciona Amazon Personalize

Amazon Personalize utiliza sus datos para entrenar modelos de recomendaciones personalizables o basados en dominios. Utilice una API de recomendaciones privada en su aplicación para solicitar recomendaciones en tiempo real. Amazon Personalize también admite flujos de trabajo por lotes para obtener recomendaciones de elementos y segmentos de usuarios.



Temas

- Detalles del flujo de trabajo de Amazon Personalize
- Términos de Amazon Personalize
- Tipos de datos que Amazon Personalize puede utilizar

Detalles del flujo de trabajo de Amazon Personalize

El flujo de trabajo de Amazon Personalize es el que se indica a continuación. Para obtener una lista de verificación que proporciona listas de las características, los requisitos y la orientación sobre datos de Amazon Personalize, consulte la Lista de verificación de preparación.

1. Correlacione el caso de uso con los recursos de Amazon Personalize: Amazon Personalize incluye recursos basados en dominio y recursos personalizados que se han configurado para distintos casos. Al correlacionar el caso de uso con un recurso de Amazon Personalize, tenga en cuenta los requisitos de datos. Después de elegir un caso de uso o una receta, esta información puede ayudarlo a preparar los datos.
2. Prepare los datos de entrenamiento: en función del caso de uso del dominio o de los requisitos de datos de una receta personalizada, prepare los datos de entrenamiento masivos en un archivo CSV. En función del caso de uso o receta, Amazon Personalize puede utilizar los datos de

interacciones de elementos, elementos, usuarios, acciones e interacciones de acciones. Si no tiene datos masivos, puede utilizar operaciones de importación individuales para recopilar datos y transmitir eventos hasta que cumpla los requisitos de entrenamiento de Amazon Personalize y los requisitos de datos del caso de uso o receta del dominio.

3. Cree archivos JSON de esquema para los datos: cree archivos JSON de esquema para cada tipo de datos que vaya a importar. Estos archivos describen la estructura y el contenido de los datos, incluidos los nombres de las columnas y los tipos de datos.
4. Cree un grupo de conjuntos de datos: un grupo de conjuntos de datos es un contenedor para los recursos de Amazon Personalize. Puede crear un grupo de conjuntos de datos de dominio con recursos preconfigurados para los dominios VIDEO_ON_DEMAND o ECOMMERCE. O puede crear un grupo de conjuntos de datos personalizados y crear solo recursos personalizados.
5. Cree esquemas y conjuntos de datos: un esquema informa a Amazon Personalize de la estructura de los datos y permite que Amazon Personalize analice los datos. Un conjunto de datos es un contenedor de datos de entrenamiento en Amazon Personalize.
6. Importe los datos de entrenamiento a conjuntos de datos: importe registros preparados de interacciones, elementos, usuarios, acciones o interacciones de acciones. Puede importar registros de forma masiva o individual.
7. Entrene e implemente un modelo: para entrenar e implementar un modelo en los dominios VIDEO_ON_DEMAND o ECOMMERCE, cree recomendaciones de dominios. En el caso de los recursos personalizados, cree una solución personalizada y una versión de solución. Para obtener recomendaciones en tiempo real, implemente la versión de la solución en una campaña.
   - Para obtener información sobre cómo crear recomendadores de dominios, consulte Recomendadores de dominios.
   - Para obtener información sobre cómo crear e implementar recursos personalizados, consulte Recursos personalizados.
8. Obtenga recomendaciones: use el recomendador o la campaña personalizada para obtener recomendaciones. Puede utilizar filtros para incluir o excluir determinados tipos de elementos de las recomendaciones. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios. Con los recursos personalizados, también puede obtener recomendaciones por lotes o segmentos de usuarios sin crear una campaña.
9. Registre los eventos en tiempo real: registre los eventos en tiempo real a medida que los clientes interactúen con las recomendaciones. De esta forma, se construyen los datos de las interacciones y se mantienen actualizados. Además, se informa a Amazon Personalize sobre los intereses actuales del usuario, lo que puede mejorar la relevancia de las recomendaciones.

Después de completar el flujo de trabajo de Amazon Personalize por primera vez, mantenga los datos actualizados y vuelva a entrenar periódicamente cualquier solución personalizada que use el entrenamiento manual. Esto permite que su modelo aprenda de la actividad más reciente de sus usuarios y mantiene y mejora la relevancia de las recomendaciones. Para obtener más información, consulte Mantenimiento de la relevancia de las recomendaciones.

Términos de Amazon Personalize

En esta sección se presentan los términos utilizados en Amazon Personalize.

Temas

- Importación y administración de datos
- Formación
- Implementación y recomendaciones del modelo

Importación y administración de datos

Los siguientes términos se refieren a la importación, exportación y formato de datos en Amazon Personalize.

conjunto de datos de acciones

Un contenedor de metadatos sobre las acciones. Una acción es una actividad que genera ingresos o interacción y que puede que quiera recomendar a los usuarios, como instalar la aplicación móvil o unirse al programa de fidelización. Los metadatos de las acciones pueden incluir la marca temporal de caducidad de la acción, el valor, los datos de frecuencia de repetición y los metadatos categóricos. Este tipo de datos los utiliza únicamente la Receta Next-Best-Action.

conjunto de datos de interacciones de acción

Un contenedor de datos históricos y en tiempo real que se recopilan de las interacciones entre los usuarios y las acciones. Cada interacción de acción consta de un ID de usuario, un ID de acción, una marca de tiempo, un tipo de evento y cualquier dato adicional sobre la interacción, como metadatos categóricos. Este tipo de datos los utiliza únicamente la Receta Next-Best-Action.

metadatos contextuales

datos de interacciones que recopila sobre el contexto de navegación de un usuario (como el dispositivo utilizado o la ubicación) cuando se produce un evento (como un clic). Los metadatos

contextuales pueden mejorar la relevancia de las recomendaciones para los usuarios nuevos y existentes.

conjunto de datos

Un contenedor para los datos que carga en Amazon Personalize. Hay cinco tipos de conjuntos de datos de Amazon Personalize: usuarios, artículos, conjunto de datos de interacciones con artículos, conjunto de datos de interacciones de acción y acciones.

grupo de conjuntos de datos

Un contenedor para los recursos de Amazon Personalize, que incluye conjuntos de datos, recomendadores de dominios y recursos personalizados. Un grupo de conjuntos de datos organiza sus recursos en colecciones independientes, de modo que los recursos de un grupo de conjuntos de datos no pueden influir en los de ningún otro grupo de conjuntos de datos. Un grupo de conjuntos de datos puede ser un grupo de conjuntos de datos de dominio o un grupo de conjuntos de datos personalizados.

Grupo de conjuntos de datos de dominio

Un grupo de conjuntos de datos que contiene recursos preconfigurados para diferentes dominios y casos de uso empresariales. Amazon Personalize gestiona el ciclo de vida de los modelos de entrenamiento y la implementación. Cuando crea un grupo de conjuntos de datos de dominio, elige su dominio comercial, importa sus datos y crea recomendadores para cada uno de sus casos de uso. Utiliza el recomendador en la aplicación para obtener recomendaciones con la operación GetRecommendations.

Si empieza con un grupo de conjuntos de datos de dominio, puede añadir recursos personalizados, como soluciones y versiones de soluciones entrenadas con recetas para casos de uso personalizados.

Grupo de conjuntos de datos personalizados

Un grupo de conjuntos de datos que contiene solo recursos personalizados, incluidas las soluciones, las versiones de las soluciones, los filtros, las campañas y los trabajos de inferencia por lotes. Utiliza una campaña para obtener recomendaciones con la operación GetRecommendations. Usted gestiona el ciclo de vida de los modelos de entrenamiento y la implementación. Si empieza con un grupo de conjuntos de datos personalizado, no podrá asociarlo a un dominio más adelante. En su lugar, cree un grupo de conjuntos de datos de dominio.

trabajo de exportación de conjuntos de datos

Herramienta de exportación de registros que envía los registros de un conjunto de datos a uno o más archivos CSV de un bucket de Amazon S3. El archivo CSV de salida incluye una fila de encabezado con nombres de columnas que coinciden con los campos del esquema del conjunto de datos.

trabajo de importación de conjuntos de datos

Una herramienta de importación masiva que rellena su conjunto de datos de Amazon Personalize con datos de un archivo CSV de su bucket de Amazon S3.

event

Una acción del usuario (como un clic, una compra o la visualización de un vídeo) que registra y carga en un conjunto de datos de interacciones de elementos de Amazon Personalize. Los eventos se importan de forma masiva desde un archivo CSV, de forma incremental con la consola de Amazon Personalize y en tiempo real.

impresiones explícitas

Una lista de elementos que agrega manualmente a un conjunto de datos de interacciones de elementos de Amazon Personalize. A diferencia de las impresiones implícitas, en las que Amazon Personalize obtiene automáticamente los datos de las recomendaciones, usted elige qué incluir en las impresiones explícitas.

impresiones implícitas

Las recomendaciones que su aplicación muestra a un usuario. A diferencia de las impresiones explícitas, que se agregan manualmente a un conjunto de datos de interacciones de elementos, Amazon Personalize obtiene automáticamente las impresiones implícitas de los datos de las recomendaciones.

datos de impresiones

La lista de elementos que presentó a un usuario cuando interactuó con un elemento concreto al hacer clic en él, verlo, comprarlo, etc. Amazon Personalize utiliza los datos de impresiones para calcular la relevancia de los nuevos artículos para un usuario en función de la frecuencia con la que los usuarios han seleccionado o ignorado el mismo artículo.

conjunto de datos de interacciones

Un contenedor de datos históricos y en tiempo real que se recopilan de las interacciones entre los usuarios y los elementos (denominados eventos). Los datos de interacciones pueden incluir datos de tipo de evento y metadatos contextuales.

conjunto de datos de elementos

Un contenedor de metadatos sobre sus elementos, como el precio, el género o la disponibilidad.

frecuencia de repetición

Tipo de metadatos de acción que se pueden importar a un conjunto de datos de acciones. Los datos de frecuencia de repetición especifican cuántos días debe esperar Amazon Personalize para recomendar una acción determinada después de la interacción de un usuario, según el historial del usuario en su conjunto de datos de interacciones de acción.

esquema

Un objeto JSON en formato Apache Avro que informa a Amazon Personalize sobre la estructura de los datos. Amazon Personalize usa su esquema para analizar los datos.

conjunto de datos de usuarios

: un contenedor de metadatos sobre sus usuarios, como la edad, el sexo o la membresía fidelizada.

Formación

Los siguientes términos hacen referencia al entrenamiento de un modelo en Amazon Personalize.

receta de similitudes entre elementos (SIMS)

Una receta de RELATED_ITEMS que utiliza los datos de un conjunto de datos de interacciones para hacer recomendaciones sobre elementos que son similares a un elemento específico. La receta de los SIMS calcula la similitud en función de la forma en que los usuarios interactúan con los elementos, en lugar de hacer coincidir los metadatos de los artículos, como el precio o el color.

item-affinity

Una receta USER_SEGMENTATION que utiliza los datos de un conjunto de datos de interacciones de elementos y un conjunto de datos de elementos para crear segmentos de usuarios para cada elemento que especifique en función de la probabilidad de que los usuarios interactúen con el elemento.

item-attribute-affinity

Una receta USER_SEGMENTATION que utiliza los datos de un conjunto de datos de interacciones de elementos y un conjunto de datos de elementos para crear un segmento de

usuarios para cada atributo de elemento que especifique en función de la probabilidad de que los usuarios interactúen con los elementos del atributo.

Receta Next-Best-Action

Esta receta genera recomendaciones en tiempo real sobre las siguientes mejores acciones para los usuarios. La siguiente mejor acción para un usuario es aquella que probablemente realice. Por ejemplo, inscribirse en su programa de fidelización, descargar su aplicación o solicitar una tarjeta de crédito. Para obtener más información, consulte Receta Next-Best-Action.

Receta Personalized-Ranking-v2

Una receta PERSONALIZED_RANKING que clasifica un conjunto de artículos que proporcione en función del nivel de interés previsto para un usuario específico. Esta receta utiliza una arquitectura basada en transformador para entrenar un modelo que aprende a partir de los datos de interacciones de elemento, los metadatos de elemento y los metadatos de usuario. Use la receta Personalized-Ranking-v2 para personalizar el orden de las listas seleccionadas de elementos o los resultados de búsqueda personalizados para un usuario específico. Puede procesar hasta cinco millones de elementos y generar recomendaciones más relevantes con una latencia menor que la versión anterior.

receta de clasificación personalizada

Una receta PERSONALIZED_RANKING que clasifica un conjunto de artículos que proporcione en función del nivel de interés previsto para un usuario específico. Use la receta personalized-ranking para personalizar el orden de las listas seleccionadas de elementos o los resultados de búsqueda personalizados para un usuario específico.

Receta Popularity-Count

Una receta USER_PERSONALIZATION que recomienda los elementos que han tenido más interacciones con usuarios únicos.

recomendador

Una herramienta de grupo de conjuntos de datos de dominio que genera recomendaciones. Crea un recomendador para un grupo de conjuntos de datos de dominio y lo utiliza en tu aplicación para obtener recomendaciones en tiempo real con la API GetRecommendations. Cuando crea un recomendador, especifica un caso de uso y Amazon Personalize entrena los modelos que lo respaldan con las mejores configuraciones para ese caso de uso.

receta

Un algoritmo de Amazon Personalize que está preconfigurado para predecir los elementos con los que interactuará un usuario (para las recetas USER_PERSONALIZATION), calcular los elementos que son similares a elementos específicos por los que un usuario ha mostrado interés (para las recetas RELATED_ITEMS) o clasificar una colección de elementos que usted proporciona en función del interés previsto para un usuario específico (para las recetas PERSONALIZED_RANKING).

solución

La receta, los parámetros personalizados y los modelos entrenados (versiones de la solución) que Amazon Personalize utiliza para generar recomendaciones.

versión de la solución

Un modelo entrenado que se crea como parte de una solución en Amazon Personalize. Implementa una versión de la solución en una campaña para activar la API de personalización que utiliza para solicitar recomendaciones.

modo de entrenamiento

El alcance del entrenamiento que se realizará al crear la versión de la solución. Hay dos modos diferentes: FULL y UPDATE. El modo FULL crea una versión de la solución completamente nueva basada en la totalidad de los datos de entrenamiento de los conjuntos de datos de su grupo de conjuntos de datos. UPDATE actualiza de forma incremental la versión de la solución existente para recomendar nuevos elementos que haya agregado desde el último entrenamiento.

> **Note**
>
> Con User-Personalization-v2, User-Personalization o Next-Best-Action, Amazon Personalize actualiza automáticamente la última versión de solución entrenada con el modo de entrenamiento FULL. Consulte Actualizaciones automáticas.

Receta User-Personalization-v2

Una receta USER_PERSONALIZATION que recomienda elementos con los que va a interactuar un usuario según las preferencias. Esta receta utiliza una arquitectura basada en transformador para entrenar un modelo que aprende a partir de los datos de interacciones de elemento, los metadatos de elemento y los metadatos de usuario. Puede procesar hasta cinco millones de

elementos y generar recomendaciones más relevantes con una latencia menor que la versión anterior.

Receta User-Personalization

Una receta USER_PERSONALIZATION basada en la red neuronal recurrente jerárquica (HRNN) que predice los elementos con los que interactuará un usuario. La receta de personalización del usuario puede utilizar los datos de exploración e impresiones de los artículos para generar recomendaciones de nuevos artículos.

Implementación y recomendaciones del modelo

Los siguientes términos se relacionan con la implementación y el uso de un modelo para generar recomendaciones.

período de optimización de acciones

El período de tiempo que Amazon Personalize utiliza al predecir las acciones que es más probable que realice el usuario. Por ejemplo, si el período de optimización de acciones es de 14 días, Amazon Personalize prevé las acciones que es más probable que los usuarios realicen en los próximos 14 días. El período de optimización de las acciones se configura al crear una solución con la Receta Next-Best-Action.

trabajo de inferencia por lotes

Una herramienta que importa los datos de entrada del lote desde un bucket de Amazon S3, utiliza la versión de la solución para generar recomendaciones y exporta las recomendaciones a un bucket de Amazon S3. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice un trabajo de inferencia por lotes para obtener recomendaciones a partir de conjuntos de datos grandes que no requieren actualizaciones en tiempo real.

trabajo de segmento por lotes

Una herramienta que importa los datos de entrada del lote desde un bucket de Amazon S3, utiliza la versión de la solución para crear segmentos de usuarios y exporta los segmentos de usuario a un bucket de Amazon S3. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice un trabajo de segmentación por lotes con una solución respaldada por la receta USER_SEGMENTATION para crear segmentos de usuarios en función de la probabilidad de que el usuario interactúe con diferentes elementos o con elementos con diferentes atributos de elemento.

campaña

Una versión de solución implementada (modelo entrenado) con capacidad de transacción dedicada y aprovisionada para crear recomendaciones en tiempo real para los usuarios de la aplicación. Después de crear una campaña, puede utilizar las operaciones de la API de `getRecommendations` o `getPersonalizedRanking` para obtener recomendaciones.

exploración de elementos

Con la exploración, las recomendaciones incluyen algunos elementos o acciones que, por lo general, es menos probable que se recomienden al usuario, como nuevos elementos o acciones, elementos o acciones con pocas interacciones, o bien elementos o acciones menos relevantes para el usuario en función de su comportamiento anterior.

atribución de métricas

Una herramienta que se utiliza para medir el impacto de las recomendaciones de elementos. Una atribución de métricas crea informes basados en los datos e interacciones de elementos que importe, así como en las métricas que especifique. Por ejemplo, la duración total de las películas vistas por los usuarios o el número total de eventos de clic.

recomendaciones

Una lista de elementos con los que Amazon Personalize predice que un usuario interactuará. Según la receta de Amazon Personalize utilizada, las recomendaciones pueden ser una lista de elementos (recetas USER_PERSONALIZATION y recetas RELATED_ITEMS) o una clasificación de una colección de elementos que haya proporcionado (recetas PERSONALIZED_RANKING).

segmentos de usuarios

Listas de usuarios con los que Amazon Personalize predice que un usuario interactuará con su catálogo. Según la receta USER_SEGMENTATION utilizada, puede crear segmentos de usuarios en función de los elementos (receta Item-Affinity) y metadatos de los elementos (receta Item-Attribute-Affinity). Los segmentos de usuarios se crean con un trabajo de segmentos por lotes.

Tipos de datos que Amazon Personalize puede utilizar

En los temas siguientes se presentan los diferentes tipos de datos que puede importar en Amazon Personalize.

Temas

- Datos de interacciones

- Datos de elementos
- Datos de usuario
- Datos de acciones
- Datos de interacciones de acción

Datos de interacciones

Una interacción es un evento que se registra y, a continuación, se importa como datos de entrenamiento. Amazon Personalize genera recomendaciones basadas principalmente en los datos de interacciones. Los datos de interacciones pueden ser los siguientes:

- Datos de valor de evento y tipo de evento
- Metadatos contextuales
- Datos de impresiones

Los datos de interacciones se importan en un conjunto de datos de interacciones de elementos. Para obtener más información sobre los conjuntos de datos de interacciones de elemento, consulte Datos de interacciones de elementos.

Datos de elementos

Entre los metadatos de los elementos que Amazon Personalize puede utilizar se incluyen los siguientes:

- Datos numéricos acerca de cada elemento, como su precio.
- Metadatos categóricos acerca de cada elemento, como el género o el color del elemento.
- Datos de marca temporal de creación para cada elemento.
- Metadatos de texto no estructurados, como descripciones de productos o sinopsis de películas.

Importa metadatos sobre sus elementos en un conjunto de datos de Elementos. Para obtener más información sobre los conjuntos de datos de Elementos, consulte Metadatos de elemento.

Datos de usuario

Entre los metadatos de usuario que Amazon Personalize puede utilizar se incluyen los siguientes:

- Datos numéricos sobre cada usuario, como su edad.
- Metadatos categóricos sobre cada usuario, como su sexo o el estado del grado de fidelización.

Los metadatos sobre los usuarios se importan en un conjunto de datos de Usuarios. Para obtener más información sobre conjuntos de datos de Usuarios, consulte Metadatos de usuario.

Datos de acciones

Entre los datos de acciones que Amazon Personalize puede utilizar se incluyen los siguientes:

- El valor empresarial o la importancia de cada acción.
- Metadatos categóricos para cada acción, como la estacionalidad o la exclusividad de la acción.
- Datos de la marca de tiempo de vencimiento de la acción que especifican cuándo debe dejar de recomendar cada acción Amazon Personalize.
- Datos de frecuencia de repetición que especifican cuánto tiempo debe esperar Amazon Personalize antes de recomendar cada acción después de la interacción de un usuario.

Debe importar los datos sobre sus acciones en un conjunto de datos de acciones. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio. Para obtener más información sobre los conjuntos de datos de acciones, consulte Metadatos de acción.

Datos de interacciones de acción

Entre los datos que Amazon Personalize puede utilizar de las interacciones de los usuarios con las acciones se incluyen los siguientes:

- Datos de tipo de evento
- Metadatos categóricos

Los datos de las interacciones se importan en un conjunto de datos de interacciones de acción. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio. Para obtener más detalles sobre los conjuntos de datos de interacciones de acción, consulte Datos de interacciones de acción.

Configuración de Amazon Personalize

Antes de utilizar Amazon Personalize, debe tener una cuenta de Amazon Web Services (AWS) con un usuario administrativo. Tras configurar los permisos necesarios, puede acceder a Amazon Personalize a través de la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) o los AWS SDK.

Temas

- Inscríbase para obtener una Cuenta de AWS
- Regiones y puntos de conexión
- Configuración de permisos
- Configurar el AWS CLI
- Configurando el AWS SDK

Inscríbase para obtener una Cuenta de AWS

Para empezar AWS, necesitas un Cuenta de AWS. Para obtener información sobre cómo crear un Cuenta de AWS, consulte Cómo empezar con un Cuenta de AWS en la Guía de AWS Account Management referencia.

Regiones y puntos de conexión

Un punto de enlace es una URL que es el punto de entrada de un servicio web. Cada punto de enlace está asociado a una región de AWS específica. Presta atención a las regiones predeterminadas de la consola Amazon Personalize y de los AWS CLI SDK de Amazon Personalize, ya que todos los componentes de Amazon Personalize de una campaña determinada (conjunto de datos, solución, campaña, seguimiento de eventos) deben crearse en la misma región. Para conocer las regiones y puntos de conexión que admite Amazon Personalize, consulte Regiones y puntos de conexión.

Configuración de permisos

Debe conceder permiso a los usuarios, grupos o roles para interactuar con los recursos de Amazon Personalize. Además, debe conceder permiso a Amazon Personalize para acceder a los recursos que cree en Amazon Personalize y para realizar tareas en su nombre.

Configuración de permisos de

1. Conceda permiso a Amazon Personalize para acceder a los recursos de Amazon Personalize y para realizar tareas en su nombre. Consulte Concesión de permiso a Amazon Personalize para que acceda a sus recursos.
2. Conceda permiso a sus usuarios, grupos o roles para interactuar con los recursos de Amazon Personalize y transfiera su rol de servicio a Amazon Personalize. Consulte Concesión de permiso a los usuarios para acceder a Amazon Personalize.
3. Modifique la política de confianza de su rol de servicio de Amazon Personalize para evitar el problema del suplente confuso. Para ver un ejemplo de política de relación de confianza, consulte Cross-service prevención confusa de diputados. Para obtener información sobre cómo modificar la política de confianza de un rol, consulte Modificación de un rol.
4. Si utiliza AWS Key Management Service (AWS KMS) para el cifrado, debe conceder permiso a Amazon Personalize y a su rol de servicio Amazon Personalize IAM para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
5. Complete los pasos de Concesión de acceso a Amazon Personalize para los recursos de Amazon S3 para usar políticas de bucket de IAM y Amazon S3 para dar a Amazon Personalize acceso a sus recursos de Amazon S3.

Temas

- Concesión de permiso a Amazon Personalize para que acceda a sus recursos
- Concesión de permiso a los usuarios para acceder a Amazon Personalize
- Concesión de acceso a Amazon Personalize para los recursos de Amazon S3
- Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave

Concesión de permiso a Amazon Personalize para que acceda a sus recursos

Para conceder permiso a Amazon Personalize para acceder a sus recursos, debe crear una política de IAM que proporcione a Amazon Personalize acceso completo a sus recursos de Amazon Personalize. O bien, puede utilizar la `AmazonPersonalizeFullAccess` política AWS gestionada. `AmazonPersonalizeFullAccess`proporciona más permisos de los necesarios. Recomendamos

crear una nueva política de IAM que solo conceda los permisos necesarios. Para obtener más información sobre las políticas administradas, consulte AWS políticas administradas.

Después de crear la política, cree un rol de IAM para Amazon Personalize y asóciele la nueva política.

Temas

- Creación de una política de IAM para Amazon Personalize
- Creación de un rol de IAM para Amazon Personalize

Creación de una política de IAM para Amazon Personalize

Cree una política de IAM que proporcione a Amazon Personalize acceso completo a sus recursos de Amazon Personalize.

Utilización del editor de política de JSON para la creación de una política

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, elija Políticas.

   Si es la primera vez que elige Políticas, aparecerá la página Welcome to Managed Policies (Bienvenido a políticas administradas). Elija Comenzar.
3. En la parte superior de la página, seleccione Crear política.
4. En la sección Editor de políticas, seleccione la opción JSON.
5. Ingrese el siguiente documento de política JSON:

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

6. Elija Siguiente.

> **Note**
>
> Puede alternar entre las opciones Visual y JSON del editor en todo momento. No obstante, si realiza cambios o selecciona Siguiente en la opción Visual del editor, es posible que IAM reestructure la política, con el fin de optimizarla para el editor visual. Para obtener más información, consulte Reestructuración de política en la Guía del usuario de IAM.

7. En la página Revisar y crear, introduzca el Nombre de la política y la Descripción (opcional) para la política que está creando. Revise los Permisos definidos en esta política para ver los permisos que concede la política.
8. Elija Crear política para guardar la nueva política.

Creación de un rol de IAM para Amazon Personalize

Para utilizar Amazon Personalize, debe crear un rol AWS Identity and Access Management de servicio para Amazon Personalize. Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM. Tras crear un rol de servicio para Amazon Personalize, conceda al rol los permisos adicionales que se indican en la lista Permisos del rol de servicio adicional según sea necesario.

Para crear el rol de servicio para Personalize (consola de IAM)

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, elija Crear rol.
3. En Tipo de entidad de confianza, elija Servicio de AWS.
4. En Servicio o caso de uso, seleccione Personalize y, a continuación, elija el caso de uso Personalize.
5. Elija Siguiente.
6. Elija la política que creó en el procedimiento anterior.

7. (Opcional) Configure un límite de permisos. Se trata de una característica avanzada que está disponible para los roles de servicio, pero no para los roles vinculados a servicios.

   a. Abra la sección Configurar límite de permisos y, a continuación, elija Utilizar un límite de permisos para controlar los permisos que puedes tener el rol como máximo.

      IAM incluye una lista de las políticas AWS gestionadas y gestionadas por los clientes de tu cuenta.

   b. Seleccione la política que desea utilizar para el límite de permisos.

8. Elija Siguiente.

9. Escriba un nombre o sufijo de nombre para el rol, que pueda ayudarle a identificar su finalidad.

   > **Important**
   >
   > Cuando asigne un nombre a un rol, tenga en cuenta lo siguiente:
   >
   > - Los nombres de los roles deben ser únicos dentro de tu perfil Cuenta de AWS y no se pueden hacer únicos por mayúsculas y minúsculas.
   >
   >   Por ejemplo, no puedes crear roles denominados tanto **`PRODROLE`** como **`prodrole`**. Cuando se utiliza un nombre de rol en una política o como parte de un ARN, el nombre de rol distingue entre mayúsculas y minúsculas, sin embargo, cuando un nombre de rol les aparece a los clientes en la consola, como por ejemplo durante el proceso de inicio de sesión, el nombre de rol no distingue entre mayúsculas y minúsculas.
   > - Dado que otras entidades podrían hacer referencia al rol, no es posible editar el nombre del rol una vez creado.

10. (Opcional) En Descripción, ingrese una descripción para el rol.

11. (Opcional) Para editar los casos de uso y los permisos de la función, en las secciones Paso 1: Seleccionar entidades confiables o en Paso 2: Agregar permisos, elija Editar.

12. (Opcional) Para ayudar a identificar, organizar o buscar el rol, agregue etiquetas como pares clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetas para obtener AWS Identity and Access Management recursos en la Guía del usuario de IAM.

13. Revise el rol y, a continuación, elija Crear rol.

Tras crear un rol para Amazon Personalize, estará listo para concederle acceso a su bucket de Amazon S3 y a cualquier AWS KMS clave.

Permisos del rol de servicio adicional

Después de crear el rol y concederle permisos para obtener acceso a sus recursos de Amazon Personalize, haga lo siguiente:

1. Modifique la política de confianza de su rol de servicio de Amazon Personalize para evitar el problema del suplente confuso. Para ver un ejemplo de política de relación de confianza, consulte Cross-service prevención confusa de diputados. Para obtener información sobre cómo modificar la política de confianza de un rol, consulte Modificación de un rol.
2. Si utiliza AWS Key Management Service (AWS KMS) para el cifrado, debe conceder permiso a Amazon Personalize y a su rol de servicio Amazon Personalize IAM para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Concesión de permiso a los usuarios para acceder a Amazon Personalize

Para proporcionar a sus usuarios acceso a Amazon Personalize, debe crear una política de IAM que conceda permiso para acceder a sus recursos de Amazon Personalize y transferir un rol de servicio a Amazon Personalize. A continuación, tendrá que usar esa política para agregar permisos a los usuarios, grupos o roles.

Creación de una política de IAM nueva para los usuarios

Cree una política de IAM que proporcione acceso completo a sus recursos de Amazon Personalize y permisos de `PassRole` para transferir su rol de servicio a Amazon Personalize (creado en Creación de un rol de IAM para Amazon Personalize).

Utilización del editor de política de JSON para la creación de una política

1. Inicia sesión en Consola de administración de AWS y abre la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, elija Políticas.

   Si es la primera vez que elige Políticas, aparecerá la página Welcome to Managed Policies (Bienvenido a políticas administradas). Elija Comenzar.
3. En la parte superior de la página, seleccione Crear política.

4. En la sección Editor de políticas, seleccione la opción JSON.
5. Ingrese el siguiente documento de política JSON:

```
{
    "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. Elija Siguiente.

> **Note**
>
> Puede alternar entre las opciones Visual y JSON del editor en todo momento. No obstante, si realiza cambios o selecciona Siguiente en la opción Visual del editor, es posible que IAM reestructure la política, con el fin de optimizarla para el editor visual. Para obtener más información, consulte Reestructuración de política en la Guía del usuario de IAM.

7. En la página Revisar y crear, introduzca el Nombre de la política y la Descripción (opcional) para la política que está creando. Revise los Permisos definidos en esta política para ver los permisos que concede la política.

8. Elija Crear política para guardar la nueva política.

Para conceder solo los permisos necesarios para llevar a cabo una tarea en Amazon Personalize, modifique la política anterior para incluir solo las acciones necesarias para el usuario. Para obtener una lista de acciones de Amazon Personalize, consulte Acciones, recursos y claves de condición de Amazon Personalize.

Dotación de acceso a Amazon Personalize

Adjunte la nueva política de IAM cuando otorgue permisos a sus usuarios.

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios gestionados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o agregue un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Concesión de acceso a Amazon Personalize para los recursos de Amazon S3

Para dar acceso a Amazon Personalize a su bucket de Amazon S3, haga lo siguiente:

1. Si aún no lo ha hecho, siga los pasos que se indican en Configuración de permisos para configurar los permisos para que Amazon Personalize pueda acceder a sus recursos de Amazon Personalize en su nombre.

2. Asocie una política al rol de servicio de Amazon Personalize (consulte Creación de un rol de IAM para Amazon Personalize) que permita el acceso a su bucket de Amazon S3. Para obtener más información, consulte Asociación de una política de Amazon S3 a su rol de servicio de Amazon Personalize.
3. Asocie una política de bucket al bucket de Amazon S3 que contiene sus archivos de datos para que Amazon Personalize pueda acceder a ellos. Para obtener más información, consulte Asociación de una política de acceso de Amazon Personalize a su bucket de Amazon S3.
4. Si utilizas AWS Key Management Service (AWS KMS) para el cifrado, debes conceder permiso a Amazon Personalize y a tu rol de servicio Amazon Personalize IAM para usar tu clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

> **Note**
>
> Como Amazon Personalize no se comunica AWS VPCs, Amazon Personalize no puede interactuar con los buckets de Amazon S3 que solo permiten el acceso a las VPC.

Temas

- Asociación de una política de Amazon S3 a su rol de servicio de Amazon Personalize
- Asociación de una política de acceso de Amazon Personalize a su bucket de Amazon S3

Asociación de una política de Amazon S3 a su rol de servicio de Amazon Personalize

Para asociar una política de Amazon S3 a su rol de Amazon Personalize, haga lo siguiente:

1. Inicie sesión en la consola de IAM (). https://console.aws.amazon.com/iam/
2. En el panel de navegación, seleccione Policies (Políticas) y seleccione Create policy (Crear política).
3. Elija la pestaña JSON y actualice la política como se indica a continuación. Reemplace `amzn-s3-demo-bucket` con el nombre de su bucket. Puede usar la siguiente política para los trabajos de importación de conjuntos de datos o los trabajos de eliminación de datos. Si utiliza un flujo de trabajo por lotes o crea un trabajo de exportación de conjuntos de datos, Amazon Personalize necesita permisos adicionales. Consulte Política de roles de servicio para flujos de trabajo por lotes o Política de buckets de Amazon S3 para exportar un conjunto de datos.

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. Elija Siguiente: etiquetas. Si lo desea, añada cualquier etiqueta y elija Revisar.
5. Asigne un nombre a la política.
6. (Opcional) En Descripción, escriba una frase breve que describa esta política, por ejemplo, **`Allow Amazon Personalize to access its Amazon S3 bucket.`**.
7. Elija Crear política.
8. En el panel de navegación, elija Roles y elija el rol que creó para Amazon Personalize. Consulte Creación de un rol de IAM para Amazon Personalize.
9. En Permisos, elija Asociar políticas.
10. Para mostrar la política en la lista, escriba parte del nombre de la política en el cuadro de filtro Filter policies (Políticas de filtro).
11. Elija la casilla situada junto a la política que ha creado anteriormente en este procedimiento.
12. Elija Asociar política.

Antes de que su rol esté listo para usarlo con Amazon Personalize, también debe asociar una política de bucket al bucket de Amazon S3 que contiene sus datos. Consulte Asociación de una política de acceso de Amazon Personalize a su bucket de Amazon S3.

Política de roles de servicio para flujos de trabajo por lotes

Para completar un flujo de trabajo por lotes, Amazon Personalize necesita permiso para acceder a su bucket de Amazon S3 y añadirle archivos. Siga los pasos anteriores para asociar la siguiente política a su rol de Amazon Personalize. Reemplace `amzn-s3-demo-bucket` con el nombre de su bucket. Para obtener más información sobre los flujos de trabajo por lotes, consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

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/*"
            ]
        }
    ]
}
```

Política de roles de servicio para exportar un conjunto de datos

Para exportar un conjunto de datos, su rol de servicio de Amazon Personalize necesita permiso para usar las acciones `PutObject` y `ListBucket` en su bucket de Amazon S3. La siguiente política de ejemplo concede los permisos de `PutObject` y `ListBucket` a Amazon Personalize. Sustituya `amzn-s3-demo-bucket` por el nombre de su bucket y asocie la política a su rol de servicio para Amazon Personalize. Para obtener información acerca de la asociación de políticas a un rol de servicio, consulte Asociación de una política de Amazon S3 a su rol de servicio de 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/*"
            ]
        }
    ]
}
```

Asociación de una política de acceso de Amazon Personalize a su bucket de Amazon S3

Amazon Personalize necesita permiso para obtener acceso al bucket de S3. Puede usar la siguiente política para los trabajos de importación de conjuntos de datos o los trabajos de eliminación de datos. Reemplace `amzn-s3-demo-bucket` con el nombre de su bucket. Para flujos de trabajo por lotes, consulte Política de buckets de Amazon S3 para flujos de trabajo por lotes.

Para obtener información acerca de las políticas de buckets de Amazon S3, consulte ¿Cómo agrego una política de bucket en 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/*"
            ]
        }
    ]
}
```

Política de buckets de Amazon S3 para flujos de trabajo por lotes

Para flujos de trabajo por lotes, Amazon Personalize necesita permiso para acceder a su bucket de Amazon S3 y añadirle archivos. Adjunte la siguiente política a su bucket. Reemplace `amzn-s3-demo-bucket` con el nombre de su bucket.

Para obtener información acerca de cómo agregar una política de bucket de Amazon S3, consulte ¿Cómo añado una política de bucket de S3? Para obtener más información sobre los flujos de trabajo por lotes, consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

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/*"
            ]
        }
    ]
}
```

Política de buckets de Amazon S3 para exportar un conjunto de datos

Para exportar un conjunto de datos, Amazon Personalize necesita permiso para usar las acciones `PutObject` y `ListBucket` en su bucket de Amazon S3. La siguiente política de ejemplo concede el principio `PutObject` y los permisos de `ListBucket` de Amazon Personalize. Sustituya `amzn-s3-demo-bucket` por el nombre de su bucket y asocie la política a su bucket. Para obtener información acerca de la edición de una política de bucket de Amazon S3, consulte ¿Cómo agrego una política de bucket de S3? en la Guía del usuario de 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/*"
            ]
        }
    ]
}
```

Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave

Si especifica una clave AWS Key Management Service (AWS KMS) cuando utiliza la consola Amazon Personalize o APIs si utiliza su AWS KMS clave para cifrar un bucket de Amazon S3, debe conceder permiso a Amazon Personalize para usar su clave. Para conceder permisos, su política de AWS KMS claves y la política de IAM asociadas a su función de servicio deben conceder a Amazon Personalize el permiso para usar su clave. Esto se aplica a la creación de lo siguiente en Amazon Personalize.

- Grupos de conjuntos de datos
- Trabajo de importación de conjuntos de datos (solo AWS KMS la política clave debe conceder permisos)
- Trabajos de exportación de conjuntos de datos
- Trabajos de inferencia por lotes
- Trabajos de segmentos por lotes
- Atribuciones de métricas

Su política AWS KMS clave y sus políticas de IAM deben conceder permisos para las siguientes acciones:

- Decrypt
- GenerateDataKey
- DescribeKey
- CreateGrant (solo se requiere en la política clave)
- ListGrants

AWS KMS La revocación de los permisos clave después de crear un recurso puede provocar problemas a la hora de crear un filtro o de obtener recomendaciones. Para obtener más información sobre AWS KMS las políticas, consulte Uso de políticas clave en AWS KMS en la Guía para AWS Key Management Service desarrolladores. Para obtener más información sobre la creación de una política de IAM, consulte Crear políticas de IAM en la Guía del usuario de IAM. Para obtener información sobre cómo adjuntar una política de IAM a un rol, consulte Incorporación y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

Temas

- Política de claves de ejemplo
- Ejemplo de política de IAM

Política de claves de ejemplo

El siguiente ejemplo de política de claves concede a Amazon Personalize y a su rol los permisos mínimos para las operaciones anteriores de Amazon Personalize. Si especifica una clave al crear un grupo de conjuntos de datos y desea exportar los datos de un conjunto de datos, su política de claves debe incluir la acción `GenerateDataKeyWithoutPlaintext`.

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": "*"
        }
    ]
}
```

Ejemplo de política de IAM

El siguiente ejemplo de política de IAM otorga a un rol los AWS KMS permisos mínimos necesarios para las operaciones anteriores de Amazon Personalize. Para los trabajos de importación de conjuntos de datos, solo la política AWS KMS clave debe conceder los permisos.

JSON

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

Configurar el AWS CLI

The AWS Command Line Interface (AWS CLI) es una herramienta de desarrollador unificada para administrar AWS servicios, incluido Amazon Personalize. Recomendamos que la instale.

1. Para instalarlo AWS CLI, siga las instrucciones de la Guía del AWS Command Line Interface usuario de la AWS Command Line Interface interfaz.
2. Para configurar AWS CLI y configurar un perfil al que llamar AWS CLI, siga las instrucciones de Configuración del AWS CLIAWS Command Line Interface manual del usuario.
3. Para confirmar que el AWS CLI perfil está configurado correctamente, ejecute el siguiente comando.

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

Si el perfil está configurado correctamente, la salida debería ser similar a la siguiente.

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

4. Para comprobar que AWS CLI está configurado para su uso con Amazon Personalize, ejecute los siguientes comandos.

```
aws personalize help
```

and

```
aws personalize-runtime help
```

and

```
aws personalize-events help
```

Si AWS CLI está configurado correctamente, verá una lista de los AWS CLI comandos compatibles con Amazon Personalize, Amazon Personalize runtime y Amazon Personalize events.

Si configuró AWS CLI y no reconoce los comandos de Amazon Personalize, actualice el AWS CLI. Para actualizar el AWS CLI, ejecuta el siguiente comando.

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

Para obtener más información, consulte Instalación de la AWS CLI utilizando pip.

Configurando el AWS SDK

Descarga e instala AWS los SDK que quieras usar. Esta guía proporciona ejemplos de SDK for Java 2.x y SDK JavaScript v3. AWS SDK para Python (Boto3) Para obtener información sobre otros AWS SDK, consulte Herramientas para Amazon Web Services. Para obtener información sobre la configuración de Amplify, consulte Documentación de Amplify.

- AWS SDK para Python (Boto3)

  Para instalar el SDK para Python (Boto3), siga las instrucciones de inicio rápido en la documentación de Boto3.

- SDK para Java 2.x

  Para obtener información sobre cómo configurar el SDK para Java 2.x, consulte el tema Introducción al SDK para Java 2.x de la Guía para desarrolladores de AWS SDK for Java 2.x .

  Para ver ejemplos de código de Amazon Personalize, consulte Ejemplos de código Java de Amazon Personalize en el repositorio de ejemplos del SDK de AWS.

- AWS SDK para JavaScript v3

  Para obtener información sobre cómo configurar el SDK para la JavaScript versión 3, consulta el tema Cómo empezar con el AWS SDK para JavaScript tema de la Guía para AWS SDK para JavaScript desarrolladores.

  Para ver ejemplos de código de Amazon Personalize, consulte Ejemplos de código de Amazon Personalize para el SDK para la versión JavaScript 3 en el repositorio de ejemplos de AWS SDK.

Tutoriales de introducción

Las secciones siguientes le ayudan a comenzar a utilizar Amazon Personalize con la consola de Amazon Personalize, AWS CLI y los SDK de AWS. Los tutoriales utilizan datos históricos que constan de 100 000 valoraciones de películas realizadas por 600 usuarios para 9700 películas.

Para simplificar los tutoriales:

- Usamos un pequeño conjunto de datos. Esto podría afectar negativamente a las métricas generadas por los recursos. Los tutoriales sirven como introducción al flujo de trabajo de Amazon Personalize y no generarán necesariamente los modelos de mayor rendimiento.
- Creamos solo un conjunto de datos de interacciones de elementos y nos basamos en el hecho de que un usuario vio una película y no en cómo la calificó. Esto simplifica la preparación de los datos de entrenamiento.
- No registramos eventos de interacción del usuario en directo. Para obtener información sobre la captura de eventos de usuario, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.

Puede elegir empezar con un grupo de conjuntos de datos de dominio o con un grupo de conjuntos de datos personalizados:

- Los grupos de conjuntos de datos de dominio proporcionan recursos optimizados para diferentes casos de uso en función de su dominio. Para empezar a crear un grupo de conjuntos de datos de dominio, complete el Requisitos previos para los ejercicios de introducción y, a continuación, complete el tutorial en Introducción a un grupo de conjuntos de datos de dominio.
- Los grupos de conjuntos de datos personalizados le permiten crear y configurar solo recursos personalizados. Para empezar a ofrecer recomendaciones de películas personalizadas a los usuarios con recursos personalizados y la receta Receta User-Personalization-v2, complete los Requisitos previos para los ejercicios de introducción y, a continuación, comience los tutoriales de Introducción a un grupo de conjuntos de datos personalizados.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Requisitos previos para los ejercicios de introducción
- Introducción a un grupo de conjuntos de datos de dominio
- Introducción a un grupo de conjuntos de datos personalizados

Requisitos previos para los ejercicios de introducción

Los siguientes pasos son requisitos previos para los ejercicios de introducción.

1. Configure permisos para que Amazon Personalize pueda acceder a sus recursos en su nombre. Esto implica crear un rol de servicio para Amazon Personalize y concederle acceso a los recursos de Amazon Personalize con una política de IAM. Para obtener más información, consulte Concesión de permiso a Amazon Personalize para que acceda a sus recursos.
2. Prepare los datos de entrenamiento y súbalos en su bucket de Amazon S3:
   - Para ver tutoriales sobre grupos de conjuntos de datos de dominios, consulte Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio).
   - Para ver tutoriales sobre grupos de conjuntos de datos personalizados, consulte Creación de los datos de entrenamiento (grupo de conjuntos de datos personalizado).
3. Conceda permiso a su rol de servicio de Amazon Personalize para acceder a sus recursos de Amazon S3, tal y como se especifica en Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio)

Para crear datos de entrenamiento, debe descargar, modificar y guardar los datos de valoraciones de películas en un bucket de Amazon Simple Storage Service (Amazon S3). A continuación, debe conceder a permiso a Amazon Personalize para leer datos del bucket.

Para crear los datos de entrenamiento

1. Descargue y descomprima el archivo zip de valoraciones de películas, ml-latest-small.zip de MovieLens, según lo recomendado para educación y desarrollo (F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872).

2. Abra el archivo `ratings.csv`. Este archivo contiene los datos de interacciones para este tutorial.

   a. Elimine la columna calificación.

   b. Cambie el nombre de las columnas `userId` y `movieId` a `USER_ID` y `ITEM_ID` respectivamente.

   c. Agregue una columna EVENT_TYPE y establezca el valor de cada registro en `watch`. Si utiliza Microsoft Excel, puede establecer el EVENT_TYPE para cada registro escribiendo `watch` en la primera celda de la columna y, a continuación, haciendo doble clic en la esquina inferior derecha de la celda. El encabezado debe ser el siguiente:

      `USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE`

      Estas columnas deben ser exactamente como se muestran para que Amazon Personalize reconozca los datos. Las primeras filas de los datos deberían tener el siguiente aspecto:

```
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
....
....
```

   Guarde el archivo `ratings.csv`.

3. Suba `ratings.csv` a su bucket de Amazon S3. Para obtener más información, consulte Subida de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.

4. Concede permiso a Amazon Personalize para leer los datos del bucket. Para obtener más información, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Creación de los datos de entrenamiento (grupo de conjuntos de datos personalizado)

Para crear datos de entrenamiento, debe descargar, modificar y guardar los datos de valoraciones de películas en un bucket de Amazon Simple Storage Service (Amazon S3). A continuación, debe conceder a permiso a Amazon Personalize para leer datos del bucket.

1. Descargue y descomprima el archivo zip de valoraciones de películas, ml-latest-small.zip de MovieLens, según lo recomendado para educación y desarrollo (F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872).
2. Abra el archivo `ratings.csv`. Este archivo contiene los datos de interacciones para este tutorial.

   a. Elimine la columna calificación.

   b. Reemplace la fila de encabezados por la siguiente:

      `USER_ID,ITEM_ID,TIMESTAMP`

      Para que Amazon Personalize reconozca los datos, estos encabezados deben coincidir exactamente con los que se muestran.

   Guarde el archivo `ratings.csv`.
3. Suba `ratings.csv` a su bucket de Amazon S3. Para obtener más información, consulte Subida de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.
4. Concede permiso a Amazon Personalize para leer los datos del bucket. Para obtener más información, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Introducción a un grupo de conjuntos de datos de dominio

En este tutorial de introducción, creará un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND, importará datos de interacciones de un archivo CSV y creará un generador de recomendaciones con el caso de uso Mejores opciones. A continuación, utilizará el generador de recomendaciones para obtener recomendaciones de películas personalizadas para un usuario.

El tutorial utiliza datos históricos que constan de 100 000 valoraciones de películas realizadas por 600 usuarios para 9700 películas.

Para empezar, complete los Requisitos previos para los ejercicios de introducción y, en función de cómo desee crear los recursos de Amazon Personalize, vaya a Introducción a un grupo de conjuntos de datos de dominio (consola), Introducción a un grupo de conjuntos de datos de dominio (SDK para Python (Boto3)), Introducción a un grupo de conjuntos de datos de dominio (SDK para Java 2.x) o Introducción a un grupo de conjuntos de datos de dominio (SDK para JavaScript v3).

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Introducción a un grupo de conjuntos de datos de dominio (consola)
- Introducción a un grupo de conjuntos de datos de dominio (SDK para Java 2.x)
- Introducción a un grupo de conjuntos de datos de dominio (SDK para Python (Boto3))
- Introducción a un grupo de conjuntos de datos de dominio (SDK para JavaScript v3)

Introducción a un grupo de conjuntos de datos de dominio (consola)

En este ejercicio, utilizará la consola de Amazon Personalize para crear un grupo de conjuntos de datos de dominio y un generador de recomendaciones de películas para un usuario determinado.

Antes de comenzar este ejercicio, revise la Requisitos previos para los ejercicios de introducción.

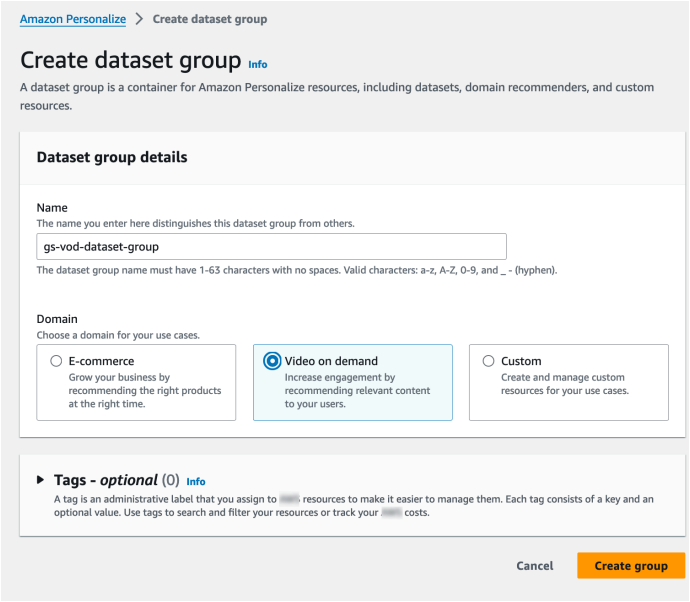
Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Paso 1: Crear un grupo de conjunto de datos de dominio

En este procedimiento, se crea un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND, se crea un conjunto de datos de interacciones de elementos con el esquema predeterminado para el dominio VIDEO_ON_DEMAND y se importan los datos de interacciones de elementos que ha creado en Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio).

Para crear un grupo de conjunto de datos de dominio

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En el panel de navegación, elija Crear grupo de conjuntos de datos.
3. En Detalles del grupo de conjuntos de datos, especifique un nombre para el grupo de conjuntos de datos.
4. En Dominio, elija Vídeo en diferido. El dominio que elija determina el esquema predeterminado que utiliza al importar datos. También determina qué casos de uso están disponibles para los generadores de recomendaciones. La pantalla debería tener un aspecto similar al siguiente.



5. Elija Crear grupo de conjuntos de datos. Aparece la página Información general. Continúe en Paso 2: Importar datos.

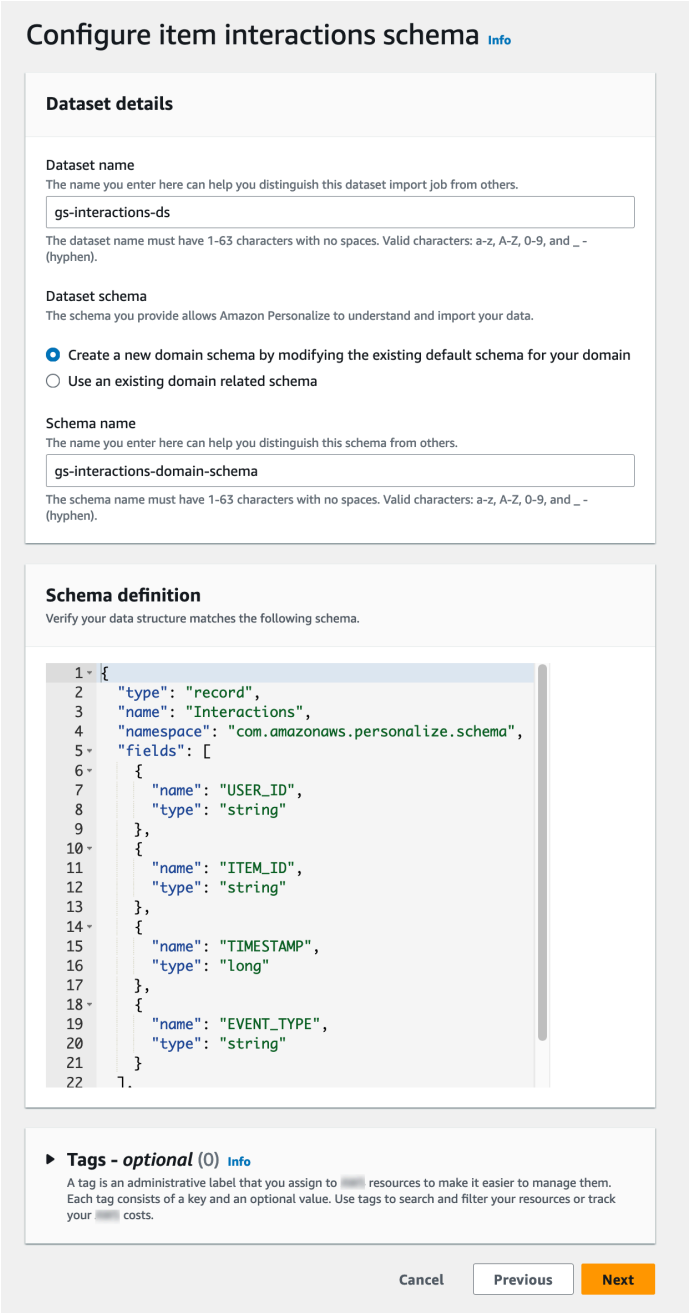
Paso 2: Importar datos

En este procedimiento, se crea un conjunto de datos de interacciones de elementos con el esquema de dominio VIDEO_ON_DEMAND predeterminado. A continuación, se importan los datos de las interacciones de elementos que ha creado en Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio).

Cómo importar datos

1. En la página Información general, en Paso 1. Cree conjuntos de datos e importe datos, elija Crear conjunto de datos y elija Conjunto de datos de interacciones de elemento.

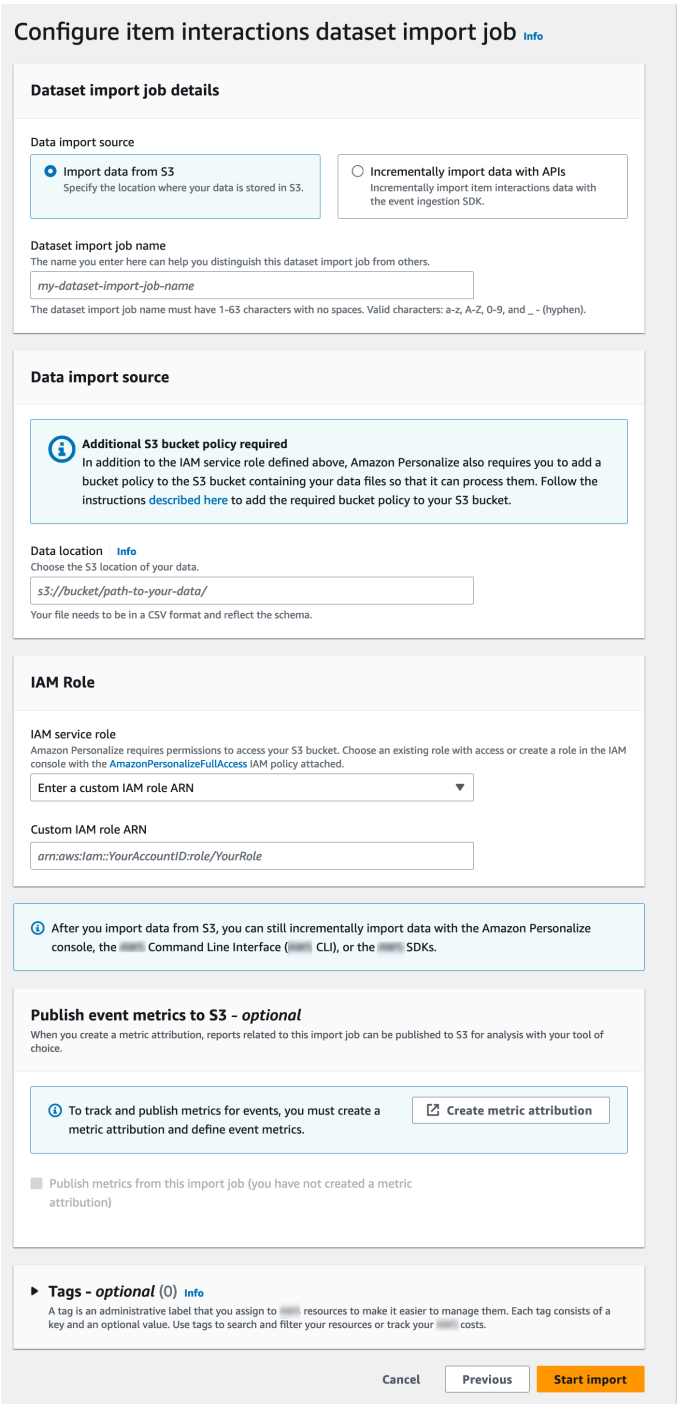
2. Elija Importar datos directamente a los conjuntos de datos de Amazon Personalize y elija Siguiente.
3. En la página Configurar esquema de interacciones de elementos, en Nombre del conjunto de datos, proporcione un nombre para su conjunto de datos de interacciones de elementos.
4. En Esquema del conjunto de datos, elija Crear un nuevo esquema de dominio modificando el esquema predeterminado existente para su dominio y especifique un nombre para el esquema. La Definición de esquema se actualiza para mostrar el esquema predeterminado del dominio VIDEO_ON_DEMAND. Deje el esquema sin cambios. La pantalla debería tener un aspecto similar al siguiente.



5. Elija Siguiente. Aparece la página para Configurar el trabajo de importación de conjuntos de datos de interacciones de elementos.
6. En la página Configurar el trabajo de importación de conjuntos de datos de interacciones de elementos, deje Origen de importación de datos sin cambios como Importar datos de S3.
7. En Nombre del trabajo de importación del conjunto de datos, especifique un nombre para el trabajo de importación.
8. En Origen de importación de datos, especifique dónde se almacenan los datos en Amazon Simple Storage Service (S3). Utilice la siguiente sintaxis:

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

9. En Rol de IAM, para Rol de servicio de IAM elija Especificar el ARN de un rol de IAM personalizado y especifique el nombre de recurso de Amazon (ARN) del rol que creó en Creación de un rol de IAM para Amazon Personalize. La pantalla debería tener un aspecto similar al siguiente.



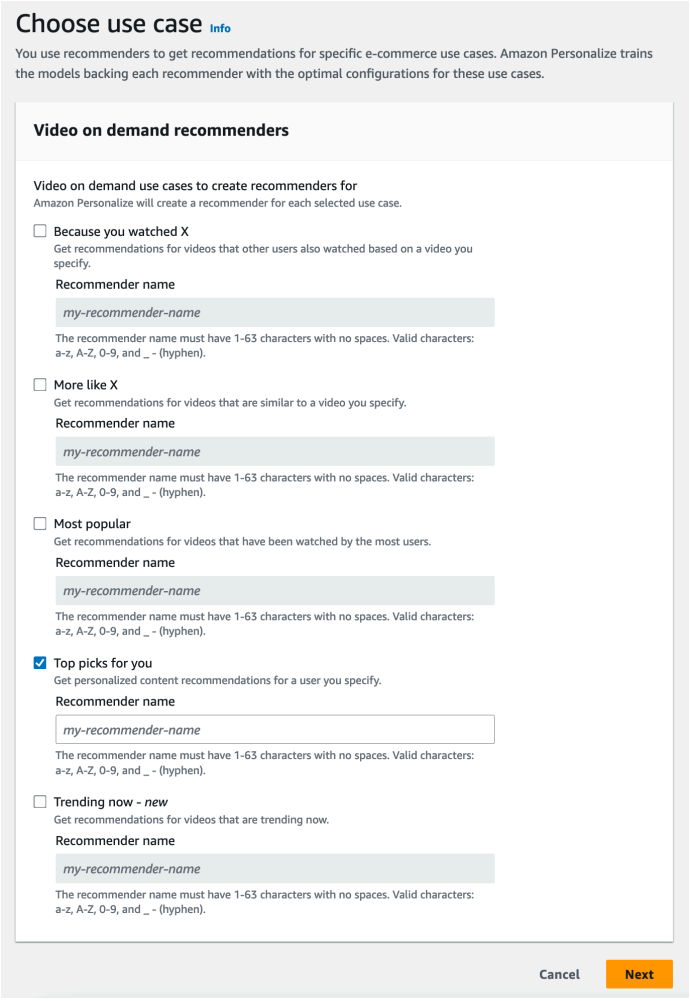
10. Elija Importar datos para importar datos. Aparece la página Información general para su grupo de conjuntos de datos de dominio. Observe el estado de la importación en la sección Configurar conjuntos de datos. Cuando el estado sea `Interaction data active`, continúe con Paso 3: Crear un generador de recomendaciones..

Paso 3: Crear un generador de recomendaciones.

En este procedimiento, crea un generador de recomendaciones para el caso de uso de Mejores opciones para el dominio de VIDEO_ON_DEMAND.

Para crear un generador de recomendaciones

1. En la página Información general de su grupo de conjuntos de datos de dominio, en el Paso 3, elija la pestaña Usar recomendadores de vídeos bajo demanda y Crear recomendadores.
2. En la página Elegir caso de uso, elija Mejores opciones y proporcione un Nombre de recomendador. La pantalla debería tener un aspecto similar al siguiente.



3. Elija Siguiente.
4. Deje los campos de la página Configuración avanzada sin cambios y elija Siguiente.
5. Revise los detalles del recomendador y seleccione Crear recomendadores para crear el recomendador.

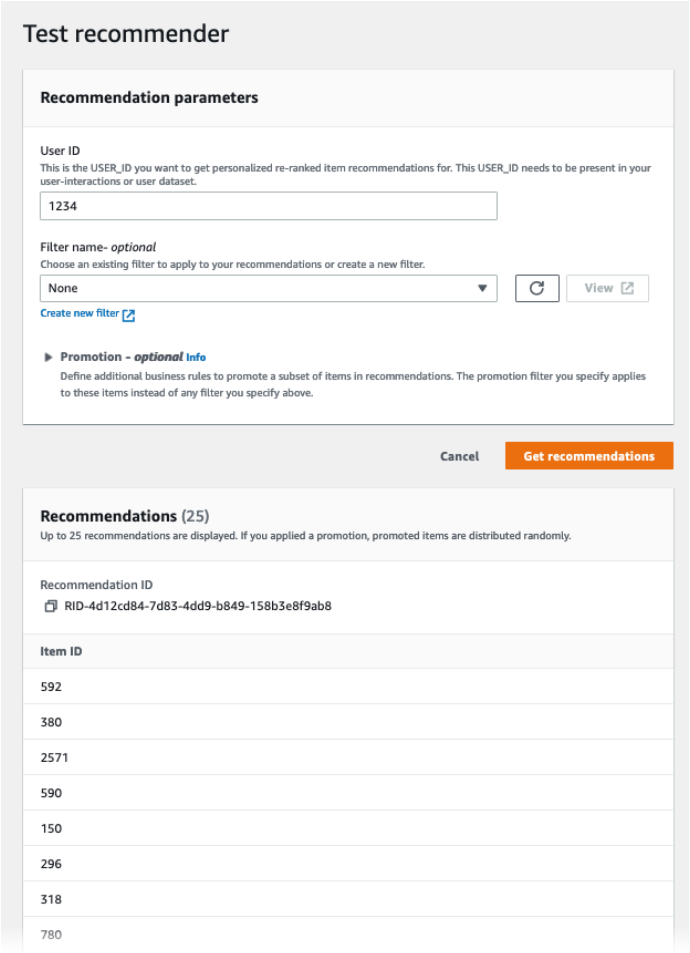
   Puede supervisar el estado de cada uno de los generadores de recomendaciones en la página Generadores de recomendaciones. Cuando el estado del recomendador sea Activo, puede usarlo para recibir recomendaciones en Paso 4: Obtener recomendaciones.

Paso 4: Obtener recomendaciones

En este procedimiento, utilizará el generador de recomendaciones que creó en el paso anterior para obtener recomendaciones.

Cómo obtener recomendaciones

1. En la página Información general de su grupo de conjuntos de datos de dominio, en el panel de navegación, elija Generador de recomendaciones.
2. En la página Generadores de recomendaciones, elija su recomendador de recomendaciones.
3. En la parte superior derecha, elija Probar.
4. En Parámetros de recomendación, especifique un ID de usuario. Deje el resto de los campos sin modificar.
5. Elija Obtener recomendaciones. Aparecerá una tabla que contiene los 25 elementos más recomendados. La pantalla debería tener un aspecto similar al siguiente.



Introducción a un grupo de conjuntos de datos de dominio (SDK para Java 2.x)

En este tutorial, se muestra cómo utilizar el SDK para Java 2.x para crear un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND. En este tutorial, se crea un recomendador para el caso de uso Mejores opciones.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Requisitos previos

Los siguientes son pasos previos para completar este tutorial:

- Complete los Requisitos previos para los ejercicios de introducción para configurar los permisos necesarios y crear los datos de entrenamiento. Si también completó Introducción a un grupo de conjuntos de datos de dominio (consola), puede reutilizar los mismos datos de origen. Si utiliza sus propios datos de origen, asegúrese de que tienen el formato que se indica en los requisitos previos.
- Configure el SDK para Java 2.x y las credenciales de AWS tal y como se especifica en el procedimiento Configuración del AWS SDK for Java 2.x de la Guía para desarrolladores de AWS SDK for Java 2.x.

Tutorial

En los siguientes pasos, configurará su proyecto para que utilice los paquetes de Amazon Personalize y cree el SDK de Amazon Personalize para clientes de Java 2.x. A continuación, se importan los datos, se crea un recomendador para el caso de uso Mejores opciones y se obtienen recomendaciones.

Paso 1: Configurar el proyecto para utilizar paquetes de Amazon Personalize

Tras completar los requisitos previos, añada las dependencias de Amazon Personalize a su archivo pom.xml e importe los paquetes de Amazon Personalize.

1. Añada las siguientes dependencias a su archivo pom.xml. Los números de versión más recientes pueden ser diferentes a los del código de ejemplo.

```
<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. Añada las siguientes instrucciones de importación al proyecto.

```
// 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;
```

Paso 2: Crear clientes de Amazon Personalize

Tras añadir las dependencias de Amazon Personalize al archivo pom.xml e importar los paquetes necesarios, cree los siguientes clientes de Amazon Personalize:

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

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

Paso 3: Importar datos

Tras inicializar sus clientes de Amazon Personalize, importe los datos históricos que creó al completar los Requisitos previos para los ejercicios de introducción. Para importar datos históricos en Amazon Personalize, haga lo siguiente:

1. Guarde el siguiente esquema de Avro como archivo JSON en su directorio de trabajo. Este esquema coincide con las columnas del archivo CSV que creó al completar el Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio).

   ```
   {
     "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. Utilice el siguiente método `createDomainSchema` para crear un esquema de dominio en Amazon Personalize. Pase lo siguiente como parámetros: un cliente del servicio de Amazon

Personalize, el nombre del esquema, `VIDEO_ON_DEMAND` para el dominio y la ruta del archivo para el esquema JSON que creó en el paso anterior. El método devuelve el nombre de recurso de Amazon (ARN) del esquema nuevo. Guárdelo para usarlo después.

```
   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. Crear un grupo de conjuntos de datos. Utilice el siguiente método `createDomainDatasetGroup` para crear un grupo de conjuntos de datos de dominio. Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, un nombre para el grupo de conjuntos de datos, y pase `VIDEO_ON_DEMAND` para el dominio. El método devuelve el ARN de su nuevo grupo de conjuntos de datos. Guárdelo para usarlo después.

```
   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. Cree un conjunto de datos de interacciones de elementos. Use el siguiente método `createDataset` para crear un conjunto de datos de interacciones de elementos. Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, el nombre de su conjunto de datos, el ARN de su esquema, el ARN de su grupo de conjuntos de datos, y `Interactions` para el tipo de conjunto de datos. El método devuelve el ARN de su nuevo conjunto de datos. Guárdelo para usarlo después.

```
   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. Importe sus datos con un trabajo de importación de conjuntos de datos. Utilice el siguiente método `createPersonalizeDatasetImportJob` para crear un trabajo de importación de conjuntos de datos.

   Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, un nombre para el trabajo y el ARN de su conjunto de datos de interacciones. Pase la ruta del bucket de Amazon S3 (`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`) donde almacenó los datos de entrenamiento y el ARN de su rol de servicio. Creó este rol como parte de los Requisitos previos para los ejercicios de introducción. El método devuelve el ARN de su trabajo de importación de conjuntos de datos. Si lo desea, guárdelo para usarlo más adelante.

```
    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 "";
    }
```

Paso 4: Crear un generador de recomendaciones

Una vez finalizado el trabajo de importación del conjunto de datos, estará listo para crear un generador de recomendaciones. Use el siguiente método `createRecommender` para crear un generador de recomendaciones. Pase lo siguiente como parámetros: un cliente de servicio de Amazon Personalize, un nombre para el generador de recomendaciones, el nombre de recurso de Amazon (ARN) de su grupo de conjuntos de datos, y `arn:aws:personalize:::recipe/aws-vod-top-picks` para el ARN de la receta. El método devuelve el ARN de su nuevo generador de recomendaciones. Guárdelo para usarlo después.

```
    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 "";
    }
```

Paso 5: Obtener recomendaciones

Después de crear un generador de recomendaciones, úselo para obtener recomendaciones. Utilice el siguiente método `getRecs` para obtener recomendaciones para un usuario. Pase como parámetros un cliente de versión ejecutable de Amazon Personalize, el nombre de recurso de Amazon (ARN) del generador de recomendaciones que ha creado en el paso anterior y un ID de usuario (por ejemplo, `123`). El método imprime la lista de elementos recomendados en la pantalla.

```
    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);
        }
    }
```

Introducción a un grupo de conjuntos de datos de dominio (SDK para Python (Boto3))

En este tutorial, se muestra cómo utilizar el SDK para Python (Boto3) para crear un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND. En este tutorial, se crea un recomendador para el caso de uso Mejores opciones.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Requisitos previos
- Tutorial
- Cómo empezar a usar Amazon Personalize APIs with Jupyter (iPython) notebooks

Requisitos previos

A continuación se indican los requisitos previos para utilizar los ejemplos de Python en esta guía:

- Complete los Requisitos previos para los ejercicios de introducción para configurar los permisos necesarios y crear los datos de entrenamiento. Si utiliza sus propios datos de origen, asegúrese de que tienen el formato que se indica en los requisitos previos.

- Configure su AWS SDK para Python (Boto3) entorno como se especifica enConfigurando el AWS SDK.

Tutorial

En los siguientes pasos, verificará su entorno y creará clientes del SDK para Python (Boto3) para Amazon Personalize. A continuación, se importan los datos, se crea un recomendador para el caso de uso Mejores opciones y se obtienen recomendaciones.

Paso 1: Verificar su entorno de Python y crear clientes boto3

Después de completar los requisitos previos, ejecute el siguiente ejemplo de Python para confirmar que su entorno está configurado correctamente. Este código también crea los clientes boto3 de Amazon Personalize que utiliza en este tutorial. Si su entorno está configurado correctamente, se muestra una lista de las recetas disponibles y puede ejecutar los demás ejemplos de este tutorial.

```
import boto3

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

response = personalize.list_recipes()

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

Paso 2: Importar datos

Tras crear los clientes boto3 de Amazon Personalize y verificar su entorno, importe los datos históricos que creó al completar los Requisitos previos para los ejercicios de introducción. Para importar datos históricos en Amazon Personalize, haga lo siguiente:

1. Utilice el siguiente código para crear un esquema en Amazon Personalize. Sustituya `gs-domain-interactions-schema` por un nombre para el esquema.

   ```
   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. Cree un grupo de conjuntos de datos con el siguiente código. Sustituya `dataset group name` por un nombre para el grupo de conjuntos de datos.

```
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. Cree un conjunto de datos de interacciones de elementos en su nuevo grupo de conjuntos de datos con el siguiente código. Asigne un nombre al conjunto de datos y proporcione el `schema_arn` y `dataset_group_arn` de los pasos anteriores.

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

dataset_arn = response['datasetArn']
```

4. Importe sus datos con un trabajo de importación de conjuntos de datos con el siguiente código. El código usa el método describe_dataset_import_job para realizar un seguimiento del estado del trabajo.

   Pase lo siguiente como parámetros: un nombre para el trabajo, el `dataset_arn` del paso anterior, la ruta del bucket de Amazon S3 (`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`) en la que almacenó los datos de entrenamiento y el ARN de su rol de servicio de IAM. Creó este rol como parte de los Requisitos previos para los ejercicios de introducción. Amazon Personalize necesita permiso para acceder al bucket. Para obtener información sobre la concesión del acceso, consulte Concesión de acceso a Amazon Personalize para los recursos de 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)
```

Paso 4: Crear un generador de recomendaciones

Una vez finalizado el trabajo de importación del conjunto de datos, estará listo para crear un generador de recomendaciones. Utilice el siguiente código para crear un generador de recomendaciones. Pase lo siguiente como parámetros: un nombre para el generador de recomendaciones, el nombre de recurso de Amazon (ARN) de su grupo de conjuntos de datos y `arn:aws:personalize:::recipe/aws-vod-top-picks` para el ARN de la receta. El código usa el método describe_recommender para realizar un seguimiento del estado del generador de recomendaciones.

```
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)
```

Paso 5: Obtener recomendaciones

Tras crear un generador de recomendaciones, se utiliza para obtener recomendaciones con el siguiente código. Pase como parámetros el nombre de recurso de Amazon (ARN) del generador de recomendaciones que ha creado en el paso anterior y un ID de usuario (por ejemplo, 123). El método imprime la lista de elementos recomendados.

```
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'])
```

Cómo empezar a usar Amazon Personalize APIs with Jupyter (iPython) notebooks

Para empezar a crear grupos de conjuntos de datos de dominio con los cuadernos de Jupyter, clone o descargue una serie de libretas que se encuentran en la carpeta notebooks_managed_domains del repositorio de muestras de Amazon Personalize. Los cuadernos le guían por la importación de datos de entrenamiento, la creación de un generador de recomendaciones y la obtención de recomendaciones con Amazon Personalize.

> **Note**
>
> Antes de empezar con los cuadernos, asegúrese de crear su entorno siguiendo los pasos del archivo README.md

Introducción a un grupo de conjuntos de datos de dominio (SDK para JavaScript v3)

En este tutorial, se muestra cómo utilizar AWS SDK para JavaScript v3 para crear un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND. En este tutorial, se crea un recomendador para el caso de uso Mejores opciones.

Para ver el código utilizado en este tutorial en GitHub, consulte Ejemplos de código de Amazon Personalize para el SDK de JavaScript v3 en el repositorio de Ejemplos de código del SDK de AWS.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Requisitos previos
- Tutorial

Requisitos previos

Los siguientes son pasos previos para completar este tutorial:

- Complete los Requisitos previos para los ejercicios de introducción para configurar los permisos necesarios y crear los datos de entrenamiento. Si también completó Introducción a un grupo de conjuntos de datos de dominio (consola), puede reutilizar los mismos datos de origen. Si utiliza sus propios datos de origen, asegúrese de que tienen el formato que se indica en los requisitos previos.
- Configure el SDK para JavaScript y las credenciales de AWS tal y como se especifica en el procedimiento Configuración del SDK para JavaScript de la Guía para desarrolladores de AWS SDK para JavaScript.

Tutorial

En los siguientes pasos, instalará las dependencias necesarias. A continuación, se crea un grupo de conjuntos de datos, se importan los datos, se crea un recomendador para el caso de uso Mejores opciones y se obtienen recomendaciones.

Si usa Node.js, puede ejecutar cada ejemplo de código guardando el ejemplo como archivo JavaScript y ejecutándolo `node <fileName.js>` a continuación.

Paso 1: Instalar las dependencias de Amazon Personalize

Después de completar los requisitos previos, instale las siguientes dependencias de Amazon Personalize:

- @aws-sdk/client-personalize
- @aws-sdk/client-personalize-runtime
- @aws-sdk/client-personalize-events (opcional para este tutorial, pero obligatorio si desea registrar eventos después de crear su generador de recomendaciones)

El siguiente es un ejemplo de un archivo `package.json` que puede usar. Para instalar las dependencias con Node.js, vaya hasta donde guardó el archivo `package.json` y ejecute `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
  }
```

```
}
```

Paso 2: Crear clientes de Amazon Personalize

Tras instalar las dependencias, cree sus clientes de Amazon Personalize. En este tutorial, en los ejemplos de código se parte del supuesto de que se crean los clientes en un archivo denominado `personalizeClients.js` almacenado en un directorio denominado `libs`.

A continuación se presenta un ejemplo de un archivo `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 };
```

Paso 3: Importar datos

Tras crear sus clientes de Amazon Personalize, importe los datos históricos que creó al completar los Requisitos previos para los ejercicios de introducción. Para importar datos históricos en Amazon Personalize, haga lo siguiente:

1. Guarde el siguiente esquema de Avro como archivo JSON en su directorio de trabajo. Este esquema coincide con las columnas del archivo CSV que creó al completar el Creación de los datos de entrenamiento (grupo de conjuntos de datos de dominio).

   ```
   {
     "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. Cree un esquema de dominio en Amazon Personalize con el siguiente código `createDomainSchema.js`. Sustituya `SCHEMA_PATH` por la ruta al archivo schema.json que acaba de crear. Actualice el `createSchemaParam` para especificar un nombre para el esquema, y para `domain` especifique `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. Cree un grupo de conjuntos de datos de dominio en Amazon Personalize con el siguiente código `createDomainDatasetGroup.js`. Actualice el `domainDatasetGroupParams` para especificar un nombre para el grupo de conjuntos de datos, y para `domain` especifique `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. Cree un conjunto de datos de interacciones de elementos en Amazon Personalize con el código `createDataset.js` siguiente. Actualice el `createDatasetParam` para especificar el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos y el esquema que acaba de crear, asigne un nombre al conjunto de datos y, para `datasetType`, especifique `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. Importe sus datos con el siguiente código `createDatasetImportJob.js`. Actualice el `datasetImportJobParam` para especificar lo siguiente:

   - Especifique un nombre para el trabajo y el ARN de su conjunto de datos de interacciones.

- Para `dataLocation`, especifique la ruta del bucket de Amazon S3 (`s3://`*`https://amzn-s3-demo-bucket.s3.region-code.amazonaws.com`*`/`*`folder name`*`/ratings.csv`) en la que almacenó los datos de entrenamiento.
- Para `roleArn`, especifique el nombre del recurso de Amazon para su rol de servicio de Amazon Personalize. Creó este rol como parte de los Requisitos previos para los ejercicios de introducción.

```
// 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();
```

Paso 4: Crear un generador de recomendaciones

Una vez finalizado el trabajo de importación del conjunto de datos, estará listo para crear un generador de recomendaciones. Para crear un generador de recomendaciones, utilice el siguiente código `createRecommender.js`. Actualice el `createRecommenderParam` con lo siguiente: especifique un nombre para el generador de recomendaciones, especifique el ARN de su grupo de conjuntos de datos y, para `recipeArn`, especifique `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();
```

Paso 5: Obtener recomendaciones

Después de crear un generador de recomendaciones, úselo para obtener recomendaciones. Utilice el siguiente código `getRecommendations.js` para obtener recomendaciones para un usuario. Actualice el `getRecommendationsParam` para especificar el ARN del generador de recomendaciones que creó en el paso anterior y especifique un ID de usuario (por ejemplo, `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();
```

Introducción a un grupo de conjuntos de datos personalizados

> **Important**
>
> En este tutorial, creará una solución que utiliza el entrenamiento automático. De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, asegúrese de eliminar la solución cuando haya terminado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Esta guía de introducción le muestra cómo ofrecer recomendaciones de películas personalizadas a los usuarios con un grupo de conjuntos de datos personalizados y la receta Receta User-

Personalization-v2. El tutorial utiliza datos históricos que constan de 100 000 valoraciones de películas realizadas por 600 usuarios para 9700 películas.

Para empezar, complete los Requisitos previos para los ejercicios de introducción y, a continuación, vaya a Introducción (consola), Introducción (AWS CLI), Introducción (SDK para Python [Boto3]) o Introducción (SDK para Java 2.x).

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Introducción (consola)
- Introducción (AWS CLI)
- Introducción (SDK para Python [Boto3])
- Introducción (SDK para Java 2.x)

Introducción (consola)

En este ejercicio, utilizará la consola de Amazon Personalize para crear un grupo de conjuntos de datos personalizados con una solución que devuelva recomendaciones de películas para un usuario determinado. Antes de comenzar este ejercicio, revise la Requisitos previos para los ejercicios de introducción.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

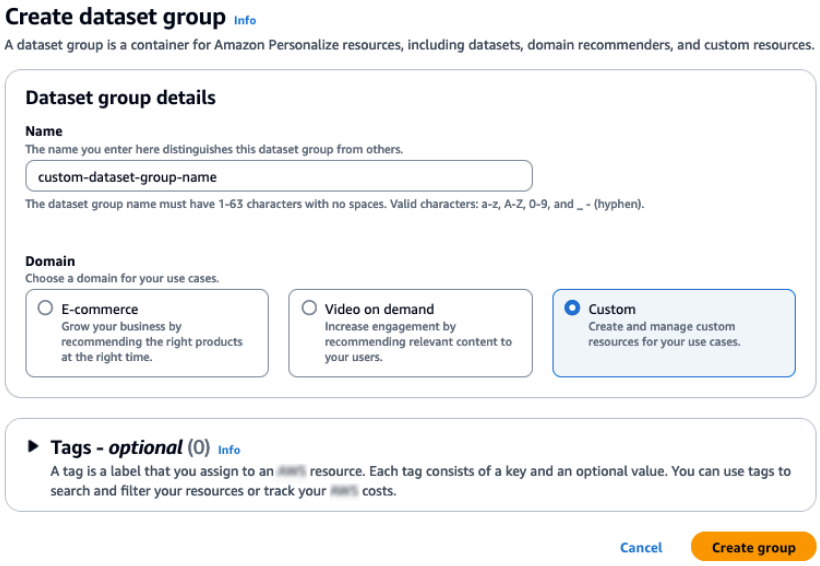
Paso 1: Crear un grupo de conjuntos de datos y un conjunto de datos

En este procedimiento, primero debe crear un grupo de conjuntos de datos. A continuación, cree un Conjunto de datos de interacciones de elementos de Amazon Personalize en el grupo de conjuntos de datos.

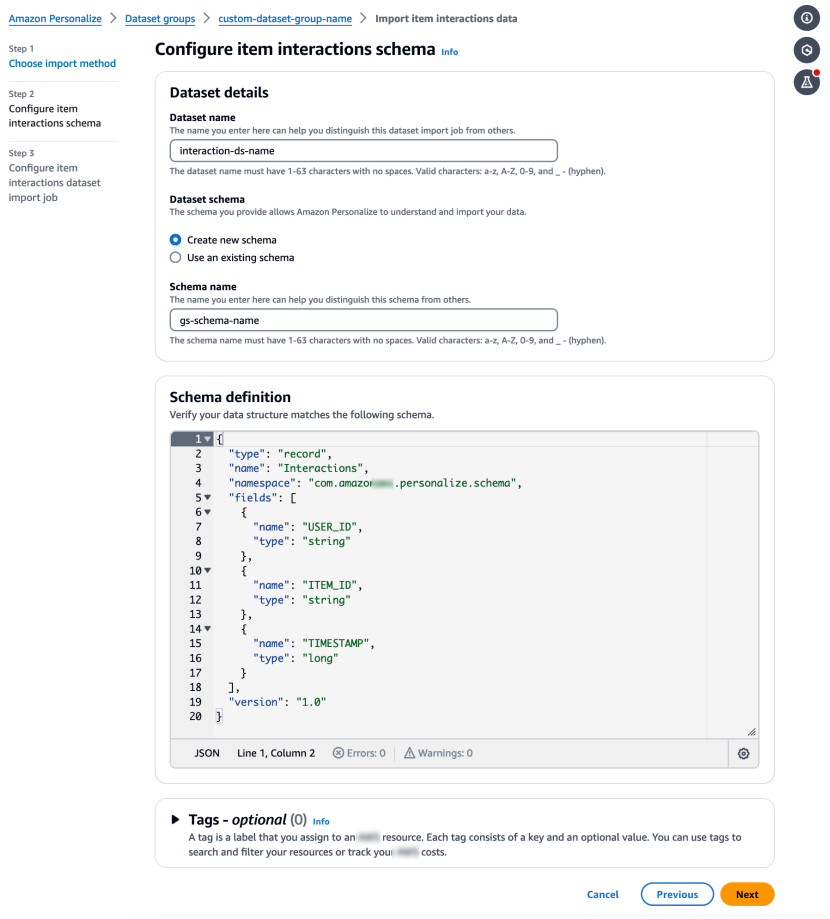
Para crear un grupo de conjunto de datos y un conjunto de datos

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.

2. Elija Crear grupo de conjuntos de datos.
3. En Detalles del grupo de conjuntos de datos, en Nombre del grupo de conjuntos de datos, especifique un nombre para el grupo de conjuntos de datos.
4. Para Dominio, elija Personalizado. La pantalla debería tener un aspecto similar al siguiente:



5. Elija Crear grupo. Aparece la página Información general.
6. En Paso 1. Cree conjuntos de datos e importe datos, elija Crear conjunto de datos y elija Conjunto de datos de interacciones de elemento.
7. Elija Importar datos directamente a los conjuntos de datos de Amazon Personalize y elija Siguiente.
8. En la página Configurar conjunto de datos de interacciones de elementos, en Nombre del conjunto de datos, especifique un nombre para su conjunto de datos.
9. Para Esquema de conjunto de datos, elija Crear esquema nuevo. En la sección Definición de esquema, se muestra un esquema de interacciones de elemento mínimo. El esquema coincide con los encabezados que añadió previamente en el archivo `ratings.csv`, por lo que no necesita realizar cambios. Si no ha creado los datos de entrenamiento, consulte Requisitos previos para los ejercicios de introducción.
10. En Nuevo nombre de esquema, especifique un nombre para el nuevo esquema. La pantalla debería tener un aspecto similar al siguiente:



11. Elija Siguiente. Aparece la página para Configurar el trabajo de importación de conjuntos de datos de interacciones de elementos. A continuación, complete Paso 2: Importar datos de interacciones de elementos para importar datos de interacciones.

Paso 2: Importar datos de interacciones de elementos

Ahora que ha creado un conjunto de datos, es el momento de importar los datos de las interacciones de elementos al conjunto de datos.

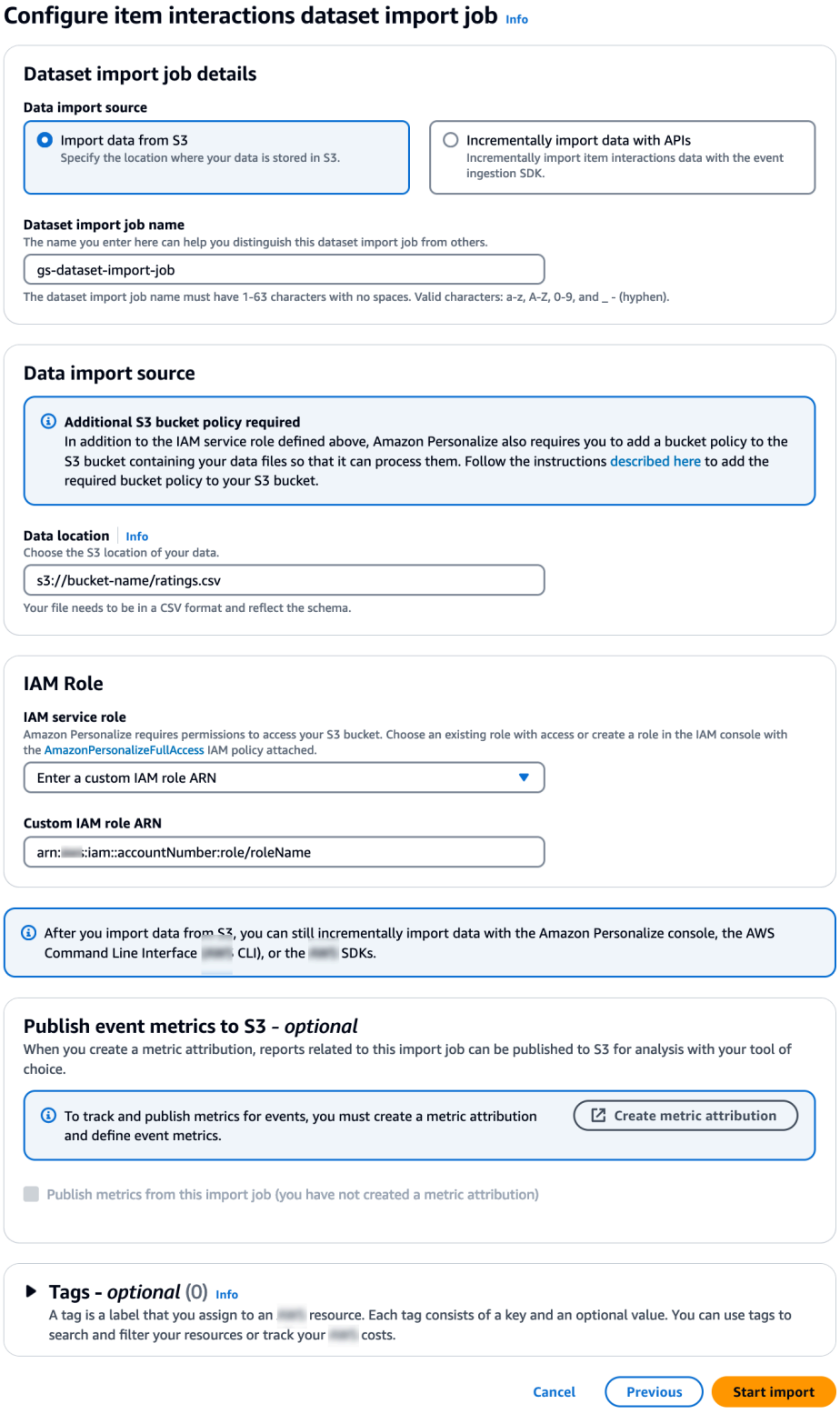
Para importar datos de interacciones de elementos

1. En la página Configurar el trabajo de importación de conjuntos de datos de interacciones de elementos, en Origen de importación de datos, elija Importar datos de S3.
2. En Nombre del trabajo de importación del conjunto de datos, especifique un nombre para su trabajo de importación.

3. En el cuadro de diálogo Política de bucket de S3 adicional requerida, si no ha concedido permisos a Amazon Personalize, siga las instrucciones para añadir la política de bucket de Amazon S3 requerida.
4. En Ubicación de datos, especifique dónde se almacenan sus datos de películas en Amazon Simple Storage Service (S3). Utilice la siguiente sintaxis:

   `s3://amzn-s3-demo-bucket/<folder path>/filename.csv`
5. En la sección Rol de IAM, para Rol de servicio de IAM, elija Especificar un ARN de rol de IAM personalizado.
6. En Custom IAM role ARN (ARN del rol de IAM personalizado), especifique el rol que creó en Creación de un rol de IAM para Amazon Personalize.

   Las secciones Detalles del trabajo de importación del conjunto de datos y de Rol de IAM deberían ser similares a las siguientes:



7. Deje las secciones Publicar métricas de eventos en S3 y Etiquetas sin cambios y elija Iniciar importación. Se inicia el trabajo de importación de los datos y aparece la página Información

general del panel. Inicialmente, el estado de Crear pendiente (seguido de Crear en curso) y el botón Crear solución está deshabilitado.

Cuando el trabajo de importación de datos ha finalizado, el estado cambia a Activo y se habilita el botón Crear solución.

8. Ahora que ha importado los datos, está listo para crear una solución en Paso 3: Crear una solución.

Paso 3: Crear una solución

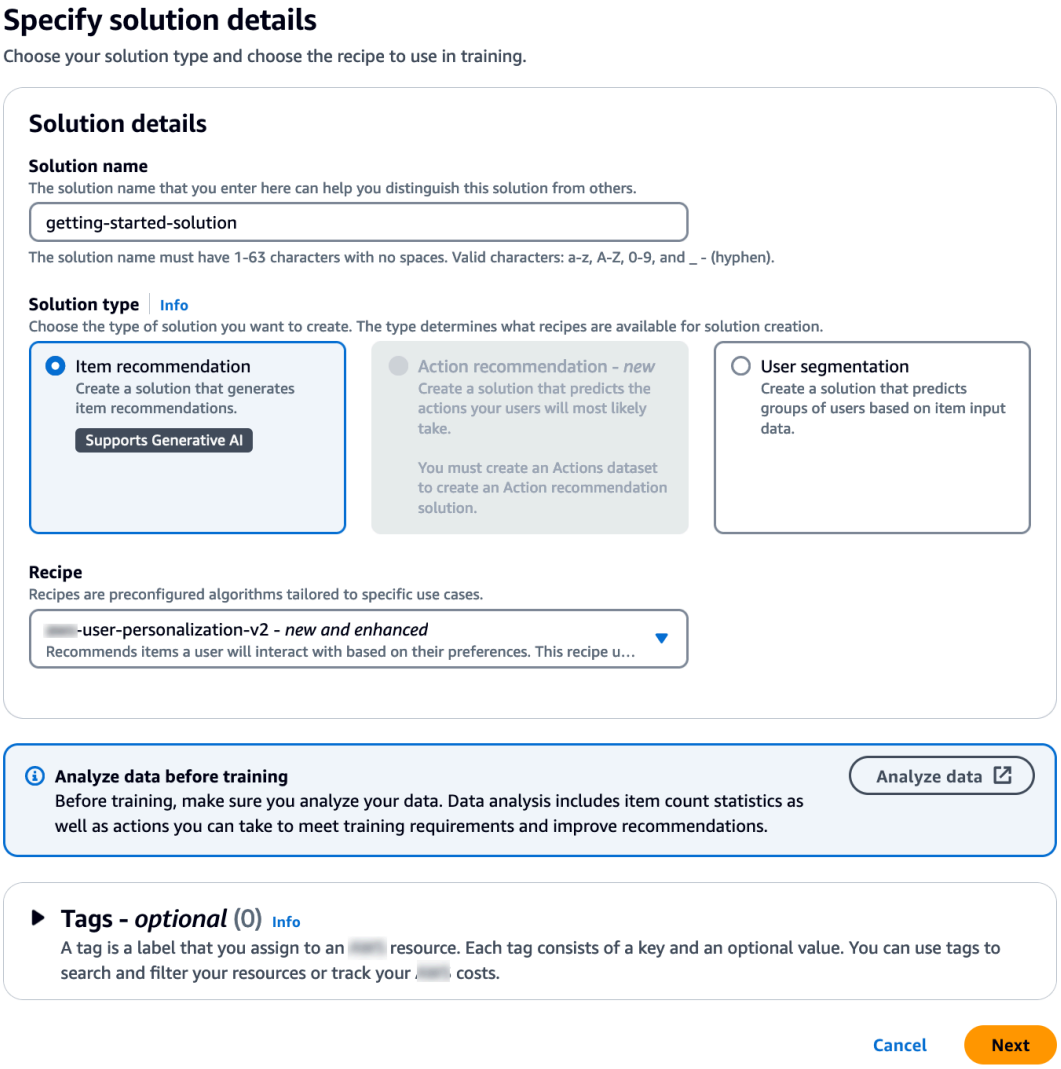
En este tutorial, se utiliza el conjunto de datos que importó en Paso 2: Importar datos de interacciones de elementos para entrenar un modelo. Los modelos entrenados se denominan versiones de solución.

> **Important**
>
> En este tutorial, creará una solución que utiliza el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, asegúrese de eliminar la solución cuando haya terminado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Para crear una solución

1. En la página Información general del grupo de conjuntos de datos, en Paso 3. Configurar los recursos de entrenamiento y recomendaciones elija Crear soluciones.
2. En Nombre de la solución, especifique un nombre para la solución.
3. En Tipo de solución, elija Recomendaciones de elementos.
4. En Receta, elija `aws-user-personalization-v2`.

   La pantalla debería tener un aspecto similar al siguiente:



5. Elija Siguiente. Deje los campos de Configuración del entrenamiento sin cambios. La solución que cree entrena automáticamente los nuevos modelos cada siete días y da más ponderación a los datos de interacción de elemento más recientes.
6. Seleccione Siguiente y revise los detalles de la solución.
7. Seleccione Crear solución y aparecerá la página de detalles de la solución. Después de crear una solución, Amazon Personalize comienza a crear la primera versión de solución en una hora. Cuando comience el entrenamiento, aparecerá en la sección Versiones de solución de la página de detalles y podrá supervisar el estado.

   Cuando el Estado de la versión de la solución sea Activo, ya puede pasar a Paso 4: Crear una campaña.

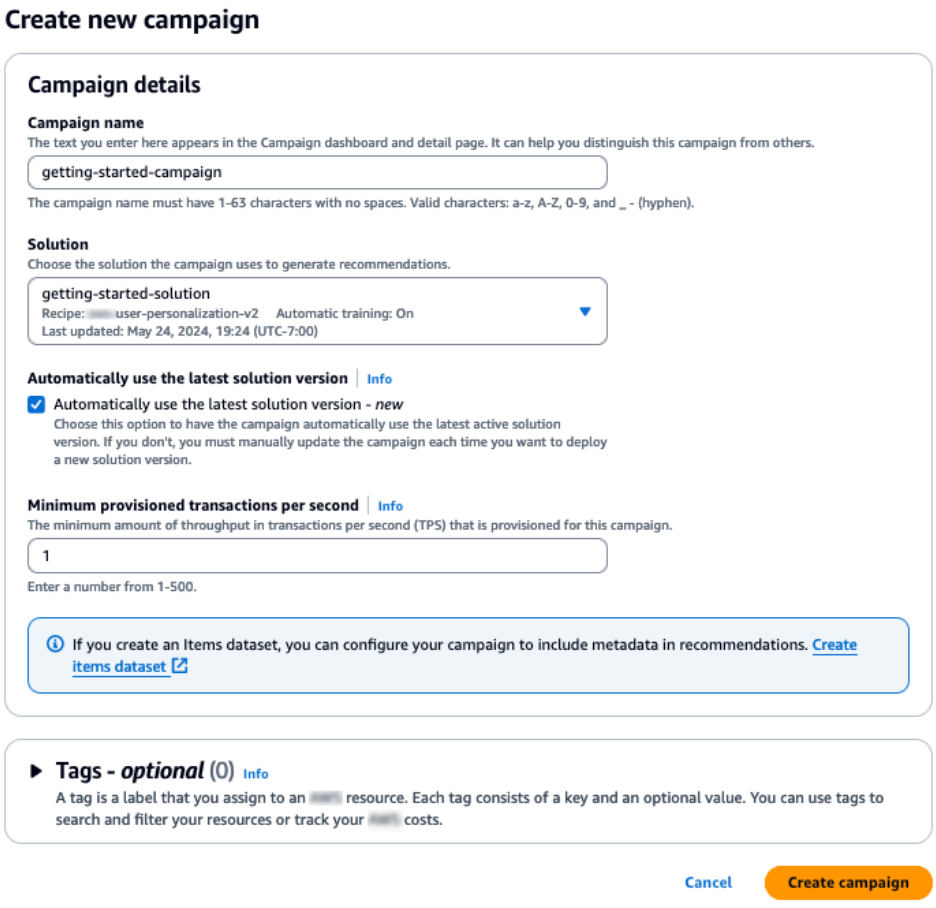
Paso 4: Crear una campaña

En este procedimiento, va a crear una campaña, la cual implementa la versión de la solución que creó en el paso anterior.

Para crear una campaña

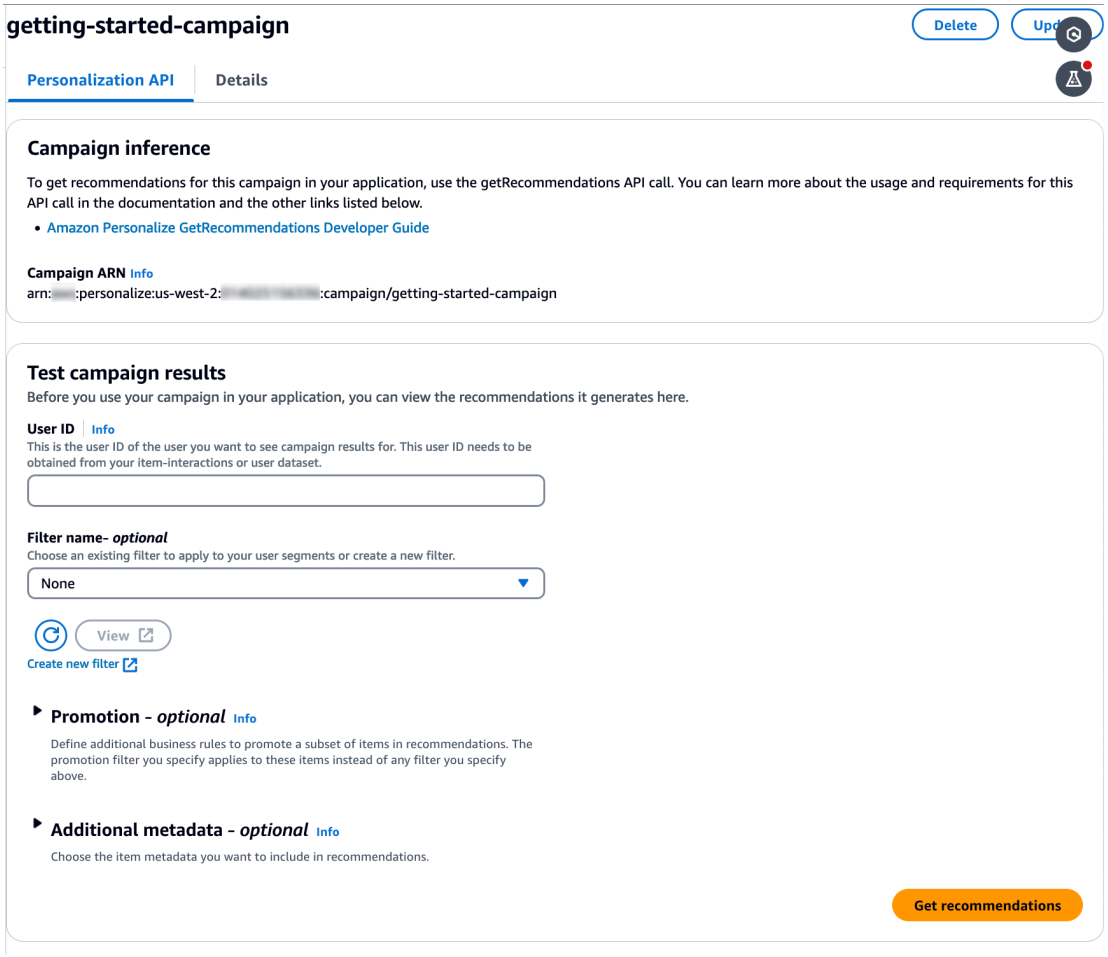
1. En el panel de navegación, expanda Recursos personalizados y elija Campañas.
2. Elija Crear una campaña. Aparece la página Crear nueva campaña.
3. En Campaign details (Detalles de la campaña), especifique un nombre para la campaña en Campaign name (Nombre de la campaña).
4. En Solución, elija la solución que ha creado en el paso anterior.
5. Seleccione Usar automáticamente la última versión de solución. Deje los demás campos sin modificar.

   La pantalla debería tener un aspecto similar al siguiente:



6. Elija Crear una campaña. Comienza la creación de la campaña y se muestran las páginas de detalles de la campaña con la sección API de personalización.

   La creación de una campaña puede tardar un par de minutos. Después de que Amazon Personalize termine de crear la campaña, la página se actualiza para mostrar la sección Probar resultados de la campaña. La pantalla debería tener un aspecto similar al siguiente:



Paso 5: Obtener recomendaciones

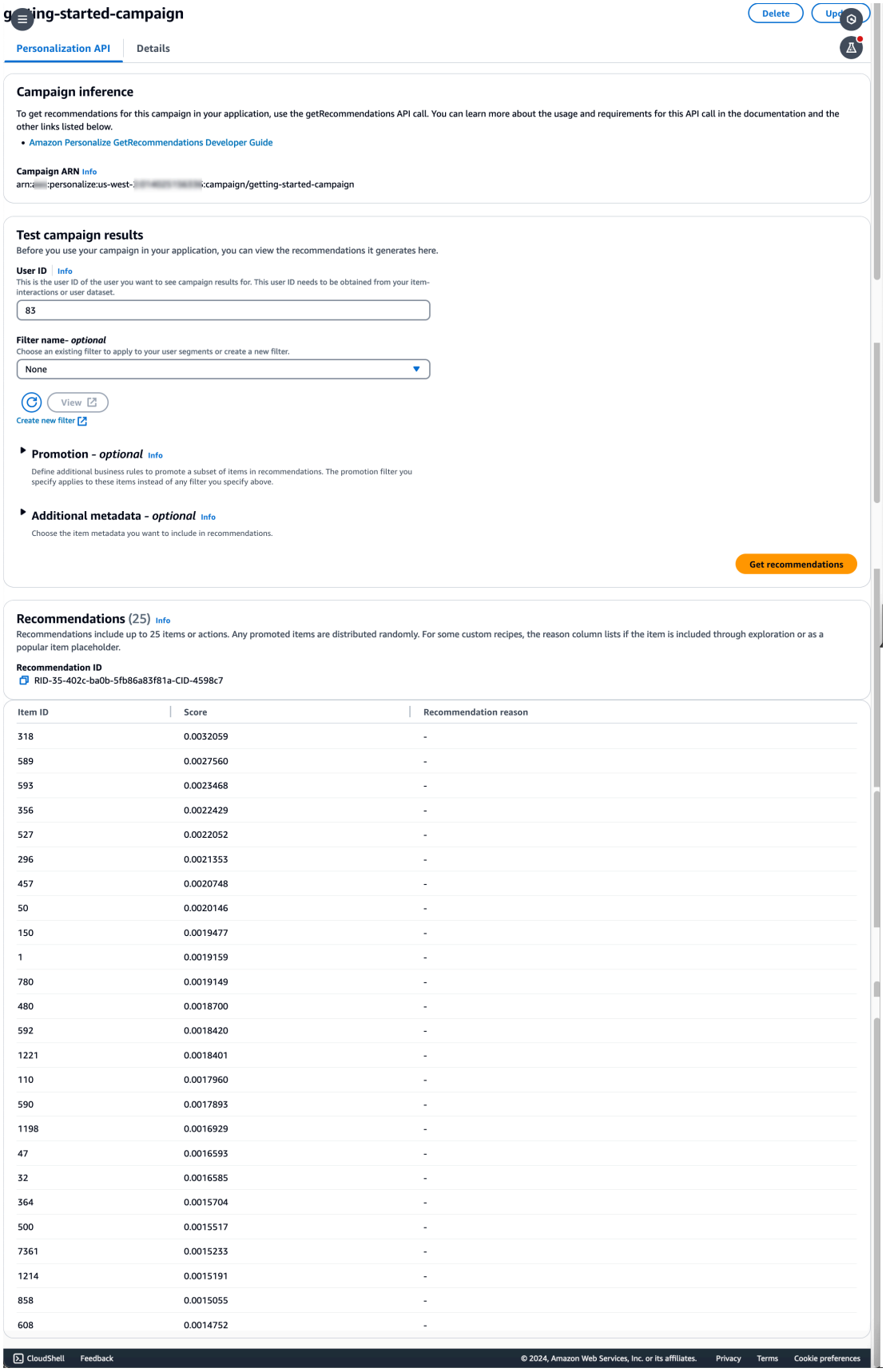
En este procedimiento, utilizará la campaña que creó en el paso anterior para obtener recomendaciones.

Cómo obtener recomendaciones

1. En Test campaign results (Probar resultado de la campaña), en User ID (ID de usuario), especifique un valor del conjunto de datos de calificaciones, por ejemplo, **83**. Deje los demás campos sin modificar.

2. Elija Obtener recomendaciones. El panel Recomendaciones muestra los ID de elementos y las puntuaciones de los elementos recomendados.

   La pantalla debería tener un aspecto similar al siguiente:


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

Introducción (AWS CLI)

En este ejercicio, utilizará la AWS Command Line Interface (AWS CLI) para explorar Amazon Personalize. Creará una campaña que devuelve recomendaciones de películas para un ID de usuario determinado.

Antes de empezar este ejercicio, haga lo siguiente:

- Consulte los de la introducción Requisitos previos para los ejercicios de introducción.
- Configure la AWS CLI, tal y como se especifica en Configurar el AWS CLI.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

> **Note**
>
> La comandos de la AWS CLI de este ejercicio se han probado en Linux. Para obtener información sobre el uso de comandos de AWS CLI en Windows, consulte Especificación de valores de parámetros para la AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.

Paso 1: Importar los datos de entrenamiento

Siga los pasos para crear un grupo de conjuntos de datos, añadir un conjunto de datos al grupo y, a continuación, rellenar el conjunto de datos con los datos de valoraciones de películas.

1. Ejecute el siguiente comando para crear un grupo de conjuntos de datos. Puede cifrar el grupo del conjunto de datos pasando un ARN de clave de AWS Key Management Service y el ARN de un rol de IAM que tenga permisos de acceso a esa clave como parámetros de entrada. Para obtener más información sobre la API, consulte 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
   ```

   Se muestra el ARN del grupo de conjuntos de datos, por ejemplo:

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

Utilice el comando `describe-dataset-group` para mostrar el grupo de conjuntos de datos que ha creado especificando el ARN del grupo de conjuntos de datos que ha obtenido.

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

Se muestran el grupo de conjuntos de datos y sus propiedades, por ejemplo:

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

> **Note**
>
> Espere hasta que el valor de `status` para el grupo de conjuntos de datos sea ACTIVE antes de poder crear un conjunto de datos en el grupo. Esta operación suele ser rápida.

Si no recuerda el ARN del grupo de conjuntos de datos, utilice el comando `list-dataset-groups` para mostrar todos los grupos de conjuntos de datos que ha creado, junto con sus ARN.

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

> **Note**
>
> Los comandos `describe-object` y `list-objects` están disponibles para la mayoría de los objetos de Amazon Personalize. Estos comandos no se muestran en el resto de este ejercicio, pero están disponibles.

2. Cree un archivo de esquema en formato JSON guardando el siguiente código en un archivo denominado `MovieRatingSchema.json`. El esquema coincide con los encabezados que añadió previamente en `ratings.csv`. El nombre del esquema es `Interactions`, que coincide con uno de los tipos de conjuntos de datos que reconoce Amazon Personalize. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de 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. Ejecute el siguiente comando para crear un esquema. Especifique el archivo que guardó en el paso anterior. En el ejemplo, el archivo pertenece a la carpeta actual. Para obtener más información sobre la API, consulte CreateSchema.

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

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

Se muestra el nombre de recurso de Amazon (ARN) del esquema, por ejemplo:

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

4. Ejecute el siguiente comando para crear un conjunto de datos vacío. Proporcione el ARN del grupo de conjuntos de datos y el ARN del esquema que obtuvo en los pasos anteriores. El valor de `dataset-type` debe coincidir con el valor de `name` del esquema del paso anterior. Para obtener más información sobre la API, consulte 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
```

Se muestra el ARN del conjunto de datos, por ejemplo:

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

5. Añada los datos de entrenamiento al conjunto de datos.

   a. Ejecute el siguiente comando para crear un trabajo de importación de conjunto de datos. Proporcione el ARN del conjunto de datos y el nombre del bucket de Amazon S3 que obtuvo en los pasos anteriores. Indique el ARN del rol de AWS Identity and Access Management (IAM) que creó en Creación de un rol de IAM para Amazon Personalize. Para obtener más información sobre la API, consulte 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
```

Se muestra el ARN del trabajo de importación de conjunto de datos, por ejemplo:

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

b. Compruebe el estado utilizando el comando `describe-dataset-import-job`. Proporcione el ARN del trabajo de importación de conjunto de datos que obtuvo en el paso anterior. Para obtener más información sobre la API, consulte DescribeDatasetImportJob.

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

Se muestran las propiedades del trabajo de importación del conjunto de datos y su estado. En un principio, el valor de `status` es CREATE PENDING; por ejemplo:

```
{
  "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
  }
}
```

La importación del conjunto de datos finaliza cuando el estado es ACTIVO. En ese momento, estará listo para entrenar el modelo utilizando el conjunto de datos especificado.

> **Note**
>
> La importación lleva tiempo. Espere hasta que se haya completado la importación del conjunto de datos para entrenar el modelo con el conjunto de datos.

Paso 2: Crear una solución (entrenar el modelo)

Para entrenar un modelo, cree la configuración para entrenarlo mediante la operación CreateSolution y deje activado el entrenamiento automático. La solución comienza automáticamente a entrenar la primera solución en una hora.

Para entrenar un modelo, se utiliza una receta y sus datos de entrenamiento. Amazon Personalize proporciona un conjunto de recetas predefinidas. Para obtener más información, consulte Elección de una receta. Para este ejercicio, debe usar la receta User-Personalization-v2.

1. Cree la configuración de entrenamiento de un modelo con el siguiente comando. Este comando crea una solución que utiliza el entrenamiento automático. Crea automáticamente una nueva versión de solución cada siete días (el valor predeterminado).

   ```
   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)\"}}"
   ```

   Se muestra el ARN de la solución, por ejemplo:

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

2. Compruebe el estado del proceso de creación con el comando `describe-solution`. Proporcione el ARN de la solución que se devolvió en el paso anterior. Para obtener más información sobre la API, consulte DescribeSolution.

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

Se muestran las propiedades de la solución, entre ellas, `status`, para la operación de creación. Por ejemplo:

```
{
  "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. Con el entrenamiento automático, el entrenamiento de versiones de solución comienza una vez que el estado de la solución sea ACTIVE. Una vez que comience el entrenamiento, puede obtener el nombre de recurso de Amazon (ARN) de la versión de solución con el siguiente comando ListSolutionVersions:

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

4. Compruebe el estado del proceso de entrenamiento de la versión de solución utilizando el comando `describe-solution-version`. Proporcione el ARN de versión de solución

que se devolvió en el paso anterior. Para obtener más información sobre la API, consulte DescribeSolutionVersion.

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

Se muestran las propiedades de la versión de solución y el valor de `status` del entrenamiento. En un principio, el estado mostrado es CREATE PENDING; por ejemplo:

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

5. Cuando la versión de la solución de `status` se muestre como ACTIVE, el entrenamiento habrá terminado.

   Ahora puede revisar las métricas de entrenamiento y crear una campaña con la versión de la solución.

   > **Note**
   >
   > El entrenamiento lleva tiempo. Espere hasta que finalice el entrenamiento (cuando el estado del entrenamiento de la versión de solución sea ACTIVE) para poder utilizar esta versión de solución en una campaña.

6. Puede validar el rendimiento de la versión de solución revisando sus métricas. Ejecute el siguiente comando para obtener las métricas de la versión de solución. Proporcione el ARN de la versión de la solución que se devolvió anteriormente. Para obtener más información sobre la API, consulte GetSolutionMetrics.

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

A continuación, se muestra una respuesta de ejemplo:

```
{
  "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
    }
}
```

Paso 3: Crear una campaña (implementar la solución)

Para poder obtener recomendaciones, debe implementar una versión de la solución. La implementación de una solución también se conoce como crear una campaña. Una vez que haya creado una campaña, su aplicación cliente puede obtener recomendaciones utilizando la API GetRecommendations.

1. Ejecute el siguiente comando para crear una campaña. Proporcione el ARN de versión de solución que se devolvió en el paso anterior. Para obtener más información sobre la API, consulte 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
   ```

   A continuación, se muestra una respuesta de ejemplo:

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

```
}
```

2. Ejecute el siguiente comando para comprobar el estado de la implementación: Proporcione el ARN de la campaña que se devolvió en el paso anterior. Para obtener más información sobre la API, consulte DescribeCampaign.

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

A continuación, se muestra una respuesta de ejemplo:

```
{
  "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**
>
> Espere hasta que la propiedad `status` se muestre como ACTIVE para poder obtener recomendaciones de la campaña.

Paso 4: Obtener recomendaciones

Ejecute el comando `get-recommendations` para obtener recomendaciones. Proporcione el ARN de la campaña que se devolvió en el paso anterior. En la solicitud, debe especificar un ID de usuario del conjunto de datos de valoraciones de películas. Para obtener más información sobre la API, consulte GetRecommendations.

> **Note**
>
> No todas las recetas admiten la API `GetRecommendations`. Para obtener más información, consulte Elección de una receta.
> El comando de la AWS CLI que se utiliza en este paso, `personalize-runtime`, no coincide con el de los pasos anteriores.

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

Como respuesta, la campaña devuelve una lista de recomendaciones de elementos (ID de películas) que podrían gustarle al usuario. La lista está ordenada por orden descendente de relevancia para el usuario.

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

Introducción (SDK para Python [Boto3])

Este tutorial le muestra cómo completar el flujo de trabajo de Amazon Personalize de principio a fin con el SDK para Python (Boto3).

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Temas

- Requisitos previos
- Tutorial
- Cómo empezar a utilizar Amazon Personalize APIs with Jupyter (iPython) notebooks

Requisitos previos

A continuación se indican los requisitos previos para utilizar los ejemplos de Python en esta guía:

- Complete los Requisitos previos para los ejercicios de introducción para configurar los permisos necesarios y crear los datos de entrenamiento. Si utiliza sus propios datos de origen, asegúrese de que tienen el formato que se indica en los requisitos previos.
- Configure su AWS SDK para Python (Boto3) entorno como se especifica enConfigurando el AWS SDK.

Tutorial

En los siguientes pasos, verificará su entorno y creará clientes del SDK para Python (Boto3) para Amazon Personalize. A continuación, importará los datos, creará e implementará una versión de la solución con una campaña, y obtendrá recomendaciones.

Paso 1: Verificar su entorno de Python y crear clientes boto3

Después de completar los requisitos previos, ejecute el siguiente ejemplo de Python para confirmar que su entorno está configurado correctamente. Este código también crea los clientes boto3 de Amazon Personalize que utiliza en este tutorial. Si su entorno está configurado correctamente, se muestra una lista de las recetas disponibles y puede ejecutar los demás ejemplos de este tutorial.

```
import boto3
```

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

response = personalize.list_recipes()

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

Paso 2: Importar datos

Tras crear los clientes boto3 de Amazon Personalize y verificar su entorno, importe los datos históricos que creó al completar los Requisitos previos para los ejercicios de introducción. Para importar datos históricos en Amazon Personalize, haga lo siguiente:

1. Utilice el siguiente código para crear un esquema en Amazon Personalize. Sustituya `getting-started-schema` por un nombre para el esquema.

```
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. Cree un grupo de conjuntos de datos con el siguiente código. Sustituya `dataset group name` por un nombre para el grupo de conjuntos de datos.

```
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. Cree un conjunto de datos de interacciones de elementos en su nuevo grupo de conjuntos de datos con el siguiente código. Asigne un nombre al conjunto de datos y proporcione el `schema_arn` y `dataset_group_arn` de los pasos anteriores.

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

dataset_arn = response['datasetArn']
```

4. Importe sus datos con un trabajo de importación de conjuntos de datos con el siguiente código. El código usa el método describe_dataset_import_job para realizar un seguimiento del estado del trabajo.

   Pase lo siguiente como parámetros: un nombre para el trabajo, el `dataset_arn` del paso anterior, la ruta del bucket de Amazon S3 (`s3://bucket name/folder name/ratings.csv`) en la que almacenó los datos de entrenamiento y el ARN de su rol de servicio de IAM. Creó este rol como parte de los Requisitos previos para los ejercicios de introducción. Amazon Personalize necesita permiso para acceder al bucket. Consulte Concesión de acceso a Amazon Personalize para los recursos de 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)
```

Paso 3: Crear una solución

Después de importar los datos, cree una solución y una versión de la solución de la siguiente manera. La solución contiene las configuraciones para entrenar un modelo y una versión de la solución es un modelo entrenado.

1. Utilice el siguiente código para crear una nueva solución. Pase lo siguiente como parámetros: el `dataset_group_arn` de antes, un nombre para la solución y el ARN para la receta User-

Personalization-v 2 ()`arn:aws:personalize:::recipe/aws-user-personalization-v2`. Guarde el ARN de su nueva solución para su uso posterior.

```
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. Cree una versión de la solución con el siguiente código. Pase como parámetro el `solution_arn` del paso anterior. El siguiente código crea una versión de la solución. Durante el entrenamiento, el código utiliza la operación DescribeSolutionVersion para recuperar el estado de la versión de la solución. Una vez finalizado el entrenamiento, el método devuelve el ARN de la nueva versión de su solución. Guárdelo para usarlo después.

```
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)
```

Paso 4: Crear una campaña

Después de crear la versión de su solución, impleméntela con una campaña de Amazon Personalize. Utilice el siguiente código para crear una campaña que implemente la versión de su solución. Pase lo siguiente como parámetros lo siguiente: el `solution_version_arn` y un nombre para la campaña. El método devuelve el nombre de recurso de Amazon (ARN) de su nueva campaña. Guárdelo para usarlo después.

```
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'])
```

Paso 5: Obtener recomendaciones

Después de crear una campaña, puede utilizarla para obtener recomendaciones. El código siguiente muestra cómo obtener recomendaciones de una campaña e imprimir el ID de cada elemento recomendado. Pase el ARN de la campaña que creó en el paso anterior. Para el ID de usuario, pase el ID de un usuario a partir de los datos de entrenamiento, por ejemplo, `123`.

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

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

Cómo empezar a utilizar Amazon Personalize APIs with Jupyter (iPython) notebooks

Para empezar a utilizar Amazon Personalize con cuadernos de Jupyter, clone o descargue una serie de cuadernos que se encuentran en la carpeta getting_started del repositorio de muestras

de Amazon Personalize. Los cuadernos le guían por la importación de datos de entrenamiento, la creación de una solución, la creación de una campaña y la obtención de recomendaciones mediante Amazon Personalize.

> **Note**
>
> Antes de empezar con los cuadernos, asegúrese de crear su entorno siguiendo los pasos del archivo README.md

Introducción (SDK para Java 2.x)

Este tutorial le muestra cómo completar el flujo de trabajo de Amazon Personalize de principio a fin con el AWS SDK for Java 2.x.

Cuando termine el ejercicio de introducción, para evitar incurrir en cargos innecesarios, elimine los recursos que ha creado. Para obtener más información, consulte Requisitos para eliminar recursos de Amazon Personalize.

Para obtener más ejemplos de , consulte Proyecto completo de Amazon Personalize.

Temas

- Requisitos previos
- Proyecto completo de Amazon Personalize

Requisitos previos

Los siguientes son pasos previos para completar este tutorial:

- Complete los Requisitos previos para los ejercicios de introducción para configurar los permisos necesarios y crear los datos de entrenamiento. Puede utilizar los mismos datos de origen que se indican en los ejercicios Introducción (consola) o Introducción (AWS CLI). Si utiliza sus propios datos de origen, asegúrese de que tienen el formato que se indica en los requisitos previos.
- Configure el SDK para Java 2.x y las credenciales de AWS tal y como se especifica en el procedimiento Configuración del AWS SDK for Java 2.x de la Guía para desarrolladores de AWS SDK for Java 2.x.

Tutorial

En los siguientes pasos, configurará su proyecto para que utilice los paquetes de Amazon Personalize y cree el SDK de Amazon Personalize para clientes de Java 2.x. A continuación, importará los datos, creará e implementará una versión de la solución con una campaña, y obtendrá recomendaciones.

Paso 1: Configurar el proyecto para utilizar paquetes de Amazon Personalize

Tras completar los requisitos previos, añada las dependencias de Amazon Personalize a su archivo pom.xml e importe los paquetes de Amazon Personalize.

1. Añada las siguientes dependencias a su archivo pom.xml. Los números de versión más recientes pueden ser diferentes a los del código de ejemplo.

```
<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. Añada las siguientes instrucciones de importación al proyecto.

```
// 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;
```

Paso 2: Crear clientes de Amazon Personalize

Tras añadir las dependencias de Amazon Personalize al archivo pom.xml e importar los paquetes necesarios, cree los siguientes clientes de Amazon Personalize:

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

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
```

```
    .region(region)
    .build();
```

Paso 3: Importar datos

Tras inicializar sus clientes de Amazon Personalize, importe los datos históricos que creó al completar los Requisitos previos para los ejercicios de introducción. Para importar datos históricos en Amazon Personalize, haga lo siguiente:

1. Guarde el siguiente esquema de Avro como archivo JSON en su directorio de trabajo. Este esquema coincide con las columnas del archivo CSV que creó al completar el Requisitos previos para los ejercicios de introducción.

```
{
  "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. Utilice el siguiente método `createSchema` para crear un esquema en Amazon Personalize. Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, el nombre del esquema, la ruta del archivo JSON de esquema que creó en el paso anterior. El método devuelve el nombre de recurso de Amazon (ARN) del esquema nuevo. Guárdelo para usarlo después.

```
    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. Crear un grupo de conjuntos de datos. Utilice el siguiente método `createDatasetGroup` para crear un grupo de conjuntos de datos. Pase como parámetros lo siguiente: un cliente del servicio de Amazon Personalize y el nombre del grupo de conjuntos de datos. El método devuelve el ARN de su nuevo grupo de conjuntos de datos. Guárdelo para usarlo después.

```
    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. Cree un conjunto de datos de interacciones de elementos. Use el siguiente método `createDataset` para crear un conjunto de datos de interacciones de elementos. Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, el nombre de su conjunto de datos, el ARN de su esquema, el ARN de su grupo de conjuntos de datos, y `Interactions` para el tipo de conjunto de datos. El método devuelve el ARN de su nuevo conjunto de datos. Guárdelo para usarlo después.

```
    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. Importe sus datos con un trabajo de importación de conjuntos de datos. Utilice el siguiente método `createPersonalizeDatasetImportJob` para crear un trabajo de importación de conjuntos de datos.

Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, un nombre para el trabajo, el ARN del conjunto de datos de interacciones de elemento, la ruta del bucket de Amazon S3 (`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`) donde ha almacenado los datos de entrenamiento y el ARN del rol de servicio (lo ha creado como parte de los Requisitos previos para los ejercicios de introducción). El método devuelve el ARN de su trabajo de importación de conjuntos de datos. Si lo desea, guárdelo para usarlo más adelante.

```
   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 "";
    }
```

Paso 4: Crear una solución

Después de importar los datos, cree una solución y una versión de la solución de la siguiente manera. La solución contiene las configuraciones para entrenar un modelo y una versión de la solución es un modelo entrenado.

1. Cree una nueva solución con el siguiente método `createPersonalizeSolution`. Pase lo siguiente como parámetros: un cliente de servicio de Amazon Personalize, el nombre de recurso de Amazon (ARN) de grupos de conjuntos de datos, un nombre para la solución y el ARN para la receta User-Personalization-v2 (`arn:aws:personalize:::recipe/aws-user-personalization-v2`). El método devuelve el ARN de su nueva solución. Guárdelo para usarlo después.

```
    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. Cree una versión de la solución con el siguiente método `createPersonalizeSolutionVersion`. Pase como parámetro el ARN de la solución del paso anterior. El siguiente código comprueba primero si la solución está lista y, a continuación, crea una versión de la solución. Durante el entrenamiento, el código utiliza la operación DescribeSolutionVersion para recuperar el estado de la versión de la solución. Una vez finalizado el entrenamiento, el método devuelve el ARN de la nueva versión de su solución. Guárdelo para usarlo después.

```
    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 "";
    }
```

Para obtener más información, consulte Creación manual de una versión de solución. Al crear una versión de la solución, puede evaluar su rendimiento antes de continuar. Para obtener más información, consulte Evaluación de una versión de solución de Amazon Personalize con métricas.

Paso 5: Crear una campaña

Después de entrenar y evaluar la versión de su solución, impleméntela con una campaña de Amazon Personalize. Use el siguiente método `createPersonalCampaign` para implementar una versión de la solución. Pase lo siguiente como parámetros: un cliente del servicio de Amazon Personalize, el nombre de recurso de Amazon (ARN) de la versión de la solución que creó en el paso anterior y un nombre para la campaña. El método devuelve el ARN de la nueva campaña. Guárdelo para usarlo después.

```
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);
    }
}
```

Para obtener más información acerca de las campañas de Amazon Personalize, consulte Implementación de una versión de solución de Amazon Personalize con una campaña.

Paso 6: Obtener recomendaciones

Después de crear una campaña, puede utilizarla para obtener recomendaciones. Utilice el siguiente método `getRecs` para obtener recomendaciones para un usuario. Pase como parámetros un cliente de versión ejecutable de Amazon Personalize, el nombre de recurso de Amazon (ARN) de la campaña que ha creado en el paso anterior y un ID de usuario (por ejemplo, `123`) a partir de los datos históricos que ha importado. El método imprime la lista de elementos recomendados en la pantalla.

```
    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);
        }
    }
```

Proyecto completo de Amazon Personalize

Para ver un proyecto integral que le muestre cómo completar el flujo de trabajo de Amazon Personalize con el SDK para Java 2.x, consulte Amazon-Personalize-Java-App en GitHub. Este proyecto incluye el entrenamiento de múltiples versiones de soluciones con diferentes recetas y el registro de eventos con la operación PutEvents.

Para ver más ejemplos, consulte el código que se encuentra en la carpeta personalize del repositorio de ejemplos del SDK de AWS.

Correlación del caso de uso con los recursos de Amazon Personalize

Las recomendaciones de Amazon Personalize pueden abordar los siguientes casos de uso:

- Generar recomendaciones personalizadas para un usuario
- Recomendar elementos similares o relacionados
- Recomendar elementos populares o de moda
- Recomendación de las siguientes mejores acciones para un usuario (solo con recursos personalizados)
- Reordenar por relevancia (solo con recursos personalizados)
- Generar segmentos de usuarios (solo con recursos personalizados)

Amazon Personalize incluye recursos basados en dominios y recursos personalizados que se han configurado para estos casos de uso. Comience por crear un grupo de conjuntos de datos de dominio o un grupo de conjuntos de datos personalizados:

- Con un grupo de conjuntos de datos de dominio, creará recursos preconfigurados y optimizados para los dominios VIDEO_ON_DEMAND o ECOMMERCE.

  Si tiene una aplicación de streaming de vídeo o de comercio electrónico, le recomendamos que comience con un grupo de conjuntos de datos de dominio. Aún puede añadir recursos personalizados, como soluciones y versiones de soluciones entrenadas para casos de uso personalizados. Además, puede seguir usando recursos personalizados para obtener recomendaciones por lotes. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio.
- Con un grupo de conjuntos de datos personalizados, puede elegir una receta que se adapte a su caso de uso. A continuación, entrenará e implementará únicamente soluciones y versiones de soluciones configurables (modelos de recomendación entrenados de Amazon Personalize). Cuando esté listo, podrá implementar la versión de solución en una campaña para obtener recomendaciones en tiempo real. O puede obtener recomendaciones por lotes sin una campaña.

Si tiene una aplicación de streaming de vídeo o de comercio electrónico, le recomendamos que cree con un grupo de conjuntos de datos personalizado. De lo contrario, comience con un grupo de conjuntos de datos de dominio y agregue recursos personalizados según sea necesario.

En las secciones siguientes se proporciona información detallada sobre los casos de uso y las recetas personalizadas disponibles en Amazon Personalize. Al correlacionar el caso de uso con un recurso de Amazon Personalize, tenga en cuenta los requisitos de datos. Después de elegir un caso de uso o una receta, esta información puede ayudarlo a preparar los datos en Preparación de los datos de entrenamiento para Amazon Personalize.

Temas

- Características del caso de uso y de la receta
- Elección de un caso de uso
- Elección de una receta

Características del caso de uso y de la receta

Con algunos casos de uso y recetas, Amazon Personalize utiliza las siguientes características para generar recomendaciones más relevantes y mejorar el descubrimiento y la fidelización de elementos.

Temas

- Personalización en tiempo real
- Exploration (Exploración)
- Actualizaciones automáticas

Personalización en tiempo real

Con algunos casos de uso y recetas, Amazon Personalize utiliza la personalización en tiempo real para actualizar y adaptar las recomendaciones en función de la evolución de los intereses de los usuarios. Actualiza las recomendaciones para un usuario a medida que registra sus interacciones con los elementos o las acciones presentes en el último entrenamiento completo. Estas interacciones se registran con un rastreador de eventos y la operación PutEvents o, en el caso de las interacciones con acciones, con la operación PutActionInteractions.

Para obtener más información sobre el registro de eventos, consulte Registro de los eventos en tiempo real para influir en las recomendaciones. Para obtener información sobre los nuevos datos que influyen en las recomendaciones en tiempo real, incluida la personalización en tiempo real, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Los siguientes casos de uso y recetas permiten la personalización en tiempo real:

- Recomendaciones (caso de uso de ECOMMERCE)
- Mejores opciones (caso de uso de VIDEO_ON_DEMAND)
- Receta User-Personalization-v2
- Receta User-Personalization
- Receta Personalized-Ranking-v2
- Receta Personalized-Ranking
- Receta Next-Best-Action

Exploration (Exploración)

Para los casos de uso de dominio y recetas personalizadas, Amazon Personalize utiliza la exploración al recomendar elementos. Con la exploración, las recomendaciones incluyen algunos elementos o acciones que, por lo general, es menos probable que se recomienden al usuario, como nuevos elementos o acciones, elementos o acciones con pocas interacciones, o bien elementos o acciones menos relevantes para el usuario en función de su comportamiento anterior. Esto mejora la visibilidad de los artículos y su fidelización cuando se trata de un catálogo que cambia rápidamente o cuando los artículos nuevos, como artículos de noticias o promociones, son más relevantes para los usuarios porque acaban de salir.

Configuración de la exploración

Si utiliza la receta User-Personalization-v2, Amazon Personalize se encarga de la configuración de la exploración y los elementos incluidos durante la exploración tienen `Reason` en `Exploration` en la respuesta de recomendación. Para asegurarse de que los elementos nuevos están incluidos en las recomendaciones, puede utilizar un filtro de promociones para promocionar elementos nuevos en función de la marca temporal de creación. Para obtener más información sobre las promociones, consulte Promoción de elementos en las recomendaciones en tiempo real.

Para el resto de casos de uso o recetas que utilizan la exploración, al crear un recomendador o una campaña personalizada, o al crear un trabajo de inferencia por lotes (recursos personalizados), puede configurar la exploración con los siguientes campos:

- Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración, las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).
- Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

  Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.

Casos de uso y recetas que utilizan la exploración

Para obtener más información acerca de cada caso de uso o receta que usa la exploración, consulte los siguientes temas:

- Recomendaciones (caso de uso de ECOMMERCE)
- Mejores opciones (caso de uso de VIDEO_ON_DEMAND)
- Receta User-Personalization-v2
- Receta User-Personalization
- Receta Next-Best-Action

Actualizaciones automáticas

Para algunos casos de uso y recetas personalizadas, Amazon Personalize actualiza automáticamente el recomendador o la versión de solución para tener en cuenta nuevos elementos o acciones para las recomendaciones. Las actualizaciones automáticas no tienen coste alguno. Para obtener una lista de casos de uso y recetas con actualizaciones automáticas, consulte Casos de uso de dominio y recetas personalizadas con actualizaciones automáticas.

Las actualizaciones automáticas funcionan de la siguiente manera:

- El momento en que Amazon Personalize actualice automáticamente la versión de solución o el recomendador depende de cómo reciba las recomendaciones:
  - Para obtener recomendaciones en tiempo real, Amazon Personalize actualiza la versión de solución o el recomendador cada dos horas.
  - En el caso de las recomendaciones de elementos por lotes, al crear un trabajo de inferencia de lotes y especificar la última versión de la solución completamente entrenada para su solución, Amazon Personalize actualiza automáticamente la versión de la solución para tener en cuenta los nuevos elementos durante la exploración. Si no especifica la versión de la solución más reciente, no se produce ninguna actualización.
- Con cada actualización, Amazon Personalize comienza a incluir nuevos elementos en las recomendaciones mediante Exploration (Exploración). Al tener en cuenta un nuevo elemento o acción, Amazon Personalize tiene en cuenta los metadatos correspondientes. Sin embargo, estos datos tendrán un mayor efecto en las recomendaciones solo después de registrar las interacciones del elemento y realizar un reentrenamiento completo.
- Para que se produzca una actualización, debe proporcionar nuevos datos de acciones, elementos o interacciones desde la última actualización automática o reentrenamiento.
- Amazon Personalize tiene en cuenta los nuevos elementos hasta que se importan 750 000 elementos. Es el número máximo de elementos que se tienen en cuenta durante el entrenamiento.

Directrices y requisitos adicionales para los recursos personalizados

Si utiliza recursos personalizados, las siguientes son pautas y requisitos para las actualizaciones automáticas:

- La versión de su solución debe implementarse en una campaña. La campaña utiliza automáticamente la versión actualizada de la solución.

- Las actualizaciones automáticas no son lo mismo que el entrenamiento automático. Una actualización automática no crea una versión de solución completamente nueva. Y el modelo no aprende a partir de los datos más recientes. Para mantener la solución, la frecuencia de entrenamiento automático debe seguir siendo de una vez a la semana como mínimo.
- Después de que la solución cree automáticamente una nueva versión de solución o usted cree una nueva, Amazon Personalize no actualizará automáticamente las versiones de solución anteriores, aunque las haya implementado en una campaña.
- Si cada dos horas no es una frecuencia suficiente, con User-Personalization puede crear manualmente una versión de solución con `trainingMode` establecido en `UPDATE` para incluir esos nuevos elementos en las recomendaciones. Recuerde que Amazon Personalize actualiza automáticamente solo la versión más reciente de la solución completamente entrenada. La versión de la solución actualizada manualmente no se actualizará automáticamente en el futuro. Si la solución utiliza el entrenamiento automático, las actualizaciones automáticas se reanudarán para la siguiente versión de solución. Si no es así, cree una nueva solución con el modo de entrenamiento establecido en `FULL` e impleméntela en una campaña.

Casos de uso de dominio y recetas personalizadas con actualizaciones automáticas

Para obtener más información acerca de cada caso de uso o receta que incluye las actualizaciones automáticas, consulte los siguientes temas:

- Recomendaciones (caso de uso de ECOMMERCE)
- Mejores opciones (caso de uso de VIDEO_ON_DEMAND)
- Receta User-Personalization-v2
- Receta User-Personalization
- Receta Next-Best-Action

Elección de un caso de uso

Al crear un generador de recomendaciones en un grupo de conjuntos de datos de dominio, debe especificar un caso de uso. Amazon Personalize entrena los modelos que respaldan al generador de recomendaciones con las mejores configuraciones para cada caso de uso. Cada dominio tiene diferentes casos de uso. Por ejemplo, si especifica VIDEO_ON_DEMAND para su grupo de conjuntos de datos de dominio, solo estarán disponibles los casos de uso de VIDEO_ON_DEMAND.

Cada caso de uso tiene requisitos diferentes para obtener recomendaciones. Algunos casos de uso requieren tipos de eventos específicos. No hay problema si incluye tipos de eventos adicionales.

Para todos los casos de uso, los datos de sus interacciones deben tener lo siguiente:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Como mínimo, 25 ID de usuarios únicos con al menos dos interacciones de elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

Temas

- Casos de uso de VIDEO_ON_DEMAND
- Casos de uso de ECOMMERCE

Casos de uso de VIDEO_ON_DEMAND

En las siguientes secciones, se enumeran los requisitos y el nombre de recurso de Amazon (ARN) para cada caso de uso de VIDEO_ON_DEMAND. Para todos los casos de uso, los datos de sus interacciones deben tener lo siguiente:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Al menos 25 usuarios únicos IDs con al menos dos interacciones entre elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

> **Note**
>
> Si usa la API de CreateRecommender, proporcione el ARN que se indica aquí para el ARN de la receta.

Temas

- Porque ha visto X
- Más como X
- Lo más popular
- Tendencia ahora
- Mejores opciones

Porque ha visto X

Obtenga recomendaciones de vídeos que otros usuarios también vieron en función del vídeo que especifique. Con este caso de uso, Amazon Personalize filtra automáticamente los vídeos que ha visto el usuario en función del userId que especifique y de los eventos `Watch`. Si aplica su propio filtro, este se aplica después de filtrar los vídeos que ha visto el usuario.

Al filtrar, Amazon Personalize tiene en cuenta como máximo 100 interacciones de elementos por usuario y por tipo de evento. Esto se aplica a cualquier filtro automático o personalizado. Puede usar la consola de Service Quotas para solicitar un aumento para este límite. Para obtener más información, consulte la sección Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

- ARN de receta: `arn:aws:personalize:::recipe/aws-vod-because-you-watched-x`
- GetRecommendations Requisitos de la API:

  `userId`: obligatorio

  `itemId`: obligatorio
- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `Watch` como mínimo.

Más como X

Obtenga recomendaciones de vídeos similares a los vídeos que especifique. Con este caso de uso, Amazon Personalize filtra automáticamente los vídeos que ha visto el usuario en función del userId

que especifique y de los eventos `Watch`. Si aplica su propio filtro, este se aplica después de filtrar los vídeos que ha visto el usuario.

Al filtrar, Amazon Personalize tiene en cuenta como máximo 100 interacciones de elementos por usuario y por tipo de evento. Esto se aplica a cualquier filtro automático o personalizado. Puede usar la consola de Service Quotas para solicitar un aumento para este límite. Para obtener más información, consulte la sección Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

- ARN de receta: `arn:aws:personalize:::recipe/aws-vod-more-like-x`
- GetRecommendations Requisitos de la API:

  `userId`: obligatorio

  `itemId`: obligatorio
- Conjuntos de datos utilizados en el entrenamiento:
  - Interacciones (obligatorio)
  - Elementos (obligatorio)
- Número obligatorio de eventos: 1000 eventos como mínimo de cualquier tipo.
- Tipos de eventos recomendados: eventos `Watch` y `Click`.

Lo más popular

Obtenga recomendaciones sobre los vídeos que han visto la mayoría de los usuarios.

- ARN de receta: `arn:aws:personalize:::recipe/aws-vod-most-popular`
- GetRecommendationsrequisitos:

  `userId`: obligatorio

  `itemId`: no se utiliza
- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `Watch` como mínimo.

Tendencia ahora

Obtenga recomendaciones sobre los vídeos que son tendencia en este momento. Los vídeos que son tendencia son elementos que se están volviendo cada vez más populares entre los usuarios. Cada dos horas, Amazon Personalize evalúa automáticamente los datos de sus interacciones e identifica los elementos que son tendencia.

- ARN de receta: `arn:aws:personalize:::recipe/aws-vod-trending-now`
- GetRecommendations Requisitos de la API:

  `userId`: Necesario solo si se filtra por CurrentUser o por elementos con los que el usuario ha interactuado

  `itemId`: no se utiliza
- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Número obligatorio de eventos: 1000 eventos como mínimo de cualquier tipo.

Mejores opciones

Obtenga recomendaciones de contenido personalizadas para el usuario que especifique. Con este caso de uso, Amazon Personalize filtra automáticamente los vídeos que ha visto el usuario en función del userId que especifique y de los eventos `Watch`. Si aplica su propio filtro, este se aplica después de filtrar los vídeos que ha visto el usuario.

Al filtrar, Amazon Personalize tiene en cuenta como máximo 100 interacciones de elementos por usuario y por tipo de evento. Esto se aplica a cualquier filtro automático o personalizado. Puede usar la consola de Service Quotas para solicitar un aumento para este límite. Para obtener más información, consulte la sección Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

Al recomendar elementos, este caso de uso utiliza real-time-personalizationy explora. Además, utiliza actualizaciones automáticas para considerar nuevos elementos como recomendaciones.

- ARN de receta: `arn:aws:personalize:::recipe/aws-vod-top-picks`
- GetRecommendations requisitos:

`userId`: obligatorio

`itemId`: no se utiliza

- Conjuntos de datos utilizados en el entrenamiento:
  - Interacciones (obligatorio)
  - Elementos (opcional)
  - Usuarios (opcional)
- Número obligatorio de eventos: 1000 eventos como mínimo.
- Tipos de eventos recomendados: eventos `Click` y `Watch`.
- Parámetros de configuración de exploración: al crear un generador de recomendaciones, puede configurar la exploración con lo siguiente.
  - Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración, las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).
  - Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

    Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.

Casos de uso de ECOMMERCE

En las siguientes secciones se enumeran los requisitos y el nombre de recurso de Amazon (ARN) para cada caso de uso de ECOMMERCE. Para todos los casos de uso, los datos de sus interacciones deben tener lo siguiente:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Al menos 25 usuarios únicos IDs con al menos dos interacciones entre elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

> **Note**
>
> Si usa la API de CreateRecommender, proporcione el ARN que se indica aquí para el ARN de la receta.

Temas

- Lo más visto
- Los más vendidos
- Con frecuencia se compran juntos
- Los clientes que vieron X también vieron
- Recomendaciones

Lo más visto

Obtenga recomendaciones de elementos populares en función del número de veces que sus clientes hayan visto un elemento.

- ARN de receta: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-views`
- GetRecommendations requisitos:

  `userId`: obligatorio

`itemId`: no se utiliza

`inputList`: N/D

- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `View` como mínimo.

Los más vendidos

Obtenga recomendaciones de elementos populares en función del número de veces que sus clientes hayan comprado un elemento.

- ARN de receta: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-purchases`
- GetRecommendations requisitos:

  `userId`: obligatorio

  `itemId`: no se utiliza

  `inputList`: N/D
- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `Purchase` como mínimo.

Con frecuencia se compran juntos

Obtenga recomendaciones de elementos que los clientes con frecuencia compren juntos con un elemento que especifique.

- ARN de receta: `arn:aws:personalize:::recipe/aws-ecomm-frequently-bought-together`
- GetRecommendations requisitos:

  `userId`: Necesario solo si filtra por CurrentUser

  `itemId`: obligatorio

`inputList`: N/D

- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `Purchase` como mínimo.

Los clientes que vieron X también vieron

Obtenga recomendaciones de elementos que los clientes también vieron en función del elemento que especifique. Con este caso de uso, Amazon Personalize filtra automáticamente los elementos que el usuario compró en función del userId que especifique y de los eventos de `Purchase`. Si aplica su propio filtro, este se aplica después de filtrar los elementos que el usuario ya ha comprado.

Al filtrar, Amazon Personalize tiene en cuenta como máximo 100 interacciones de elementos por usuario y por tipo de evento. Esto se aplica a cualquier filtro automático o personalizado. Puede usar la consola de Service Quotas para solicitar un aumento para este límite. Para obtener más información, consulte la sección Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

- ARN de receta: `arn:aws:personalize:::recipe/aws-ecomm-customers-who-viewed-x-also-viewed`
- GetRecommendations requisitos:

  `userId`: obligatorio

  `itemId`: obligatorio

  `inputList`: N/D
- Conjuntos de datos utilizados durante el entrenamiento: solo conjunto de datos de interacciones de elementos (obligatorio)
- Tipos de eventos obligatorios: 1000 eventos de `View` como mínimo.
- Tipos de eventos recomendados: eventos de `Purchase`.

Recomendaciones

Obtenga recomendaciones personalizadas de elementos en función del usuario que especifique. Con este caso de uso, Amazon Personalize filtra automáticamente los elementos que el usuario ha comprado en función del userId que especifique y de los eventos de `Purchase`. Si aplica su propio filtro, este se aplica después de filtrar los elementos que el usuario ya ha comprado.

Al filtrar, Amazon Personalize tiene en cuenta como máximo 100 interacciones de elementos por usuario y por tipo de evento. Esto se aplica a cualquier filtro automático o personalizado. Puede usar la consola de Service Quotas para solicitar un aumento para este límite. Para obtener más información, consulte la sección Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

Al recomendar elementos, este caso de uso utiliza real-time-personalizationy explora. Además, utiliza actualizaciones automáticas para considerar nuevos elementos como recomendaciones.

- ARN de receta: `arn:aws:personalize:::recipe/aws-ecomm-recommended-for-you`
- GetRecommendations requisitos:

  `userId`: obligatorio

  `itemId`: no se utiliza

  `inputList`: N/D
- Conjuntos de datos utilizados en el entrenamiento:
  - Interacciones (obligatorio)
  - Elementos (opcional)
  - Usuarios (opcional)
- Número obligatorio de eventos: 1000 eventos como mínimo.
- Tipos de eventos recomendados: eventos `View` y `Purchase`.
- Parámetros de configuración de exploración: al crear un generador de recomendaciones, puede configurar la exploración con lo siguiente.
  - Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración,

las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).

- Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

  Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.

Elección de una receta

Al crear una solución personalizada, especifique una receta y configure los parámetros de entrenamiento. Las recetas son algoritmos de Amazon Personalize preparados para casos de uso específicos. Amazon Personalize proporciona recetas, basadas en casos de uso comunes, para modelos de entrenamiento. Al crear una versión de solución para la solución, Amazon Personalize entrena los modelos que respaldan la versión de la solución en función de la receta y la configuración de entrenamiento.

Las recetas de Amazon Personalize utilizan lo siguiente durante el entrenamiento:

- Atributos predefinidos de los datos
- Transformaciones de características predefinidas
- Algoritmos predefinidos
- Ajustes de parámetros iniciales para los algoritmos

Para optimizar el modelo, puede anular muchos de estos parámetros cuando cree una solución. Para obtener más información, consulte Hiperparámetros y HPO.

Temas

- Tipos de recetas por caso de uso de Amazon Personalize
- Recetas de Amazon Personalize
- Visualización de las recetas de Amazon Personalize disponibles
- Receta User-Personalization-v2
- Receta User-Personalization
- Receta Trending-Now
- Receta Popularity-Count
- Receta Personalized-Ranking-v2
- Receta Personalized-Ranking
- Receta de similitud semántica
- Receta Similar-Items
- Receta SIMS
- Receta Next-Best-Action
- Receta Item-Affinity
- Receta Item-Attribute-Affinity
- Recetas de HRNN heredadas

Tipos de recetas por caso de uso de Amazon Personalize

Para elegir su receta, elija primero su caso de uso en la siguiente lista y tome nota del tipo de receta correspondiente.

- Recomendación de elementos para usuarios (recetas de USER_PERSONALIZATION)

  Para ofrecer recomendaciones personalizadas para sus usuarios, entrene su modelo con una receta de USER_PERSONALIZATION. Las recomendaciones personalizadas ayudan a impulsar una mejor participación y conversión.
- Clasificación de los elementos para un usuario (recetas PERSONALIZED_RANKING)

  Para personalizar el orden de las listas seleccionadas o los resultados de búsqueda para sus usuarios, entrene su modelo con una receta PERSONALIZED_RANKING. Las recetas PERSONALIZED_RANKING crean una lista personalizada al volver a clasificar una colección de

elementos de entrada en función del nivel de interés previsto para un usuario determinado. Las listas personalizadas mejoran la experiencia del cliente y aumentan su fidelidad y compromiso.

- Recomendación de tendencias o elementos populares (recetas POPULAR_ITEMS)

  Para recomendar tendencias o elementos populares, use una receta POPULAR_ITEMS. Puede usar POPULAR_ITEMS si sus clientes le dan un gran valor al material con el que interactúan otros usuarios. Entre los usos habituales se incluyen la recomendación contenido viral en las redes sociales, elementos de noticias de última hora o vídeos deportivos recientes.

- Recomendación de elementos similares (recetas RELATED_ITEMS)

  Para recomendar elementos similares, como elementos que se compran juntos con frecuencia o películas que otros usuarios también han visto, debe usar una receta RELATED_ITEMS. La recomendación de elementos similares puede ayudar a sus clientes a descubrirlos y puede aumentar la tasa de conversión de los usuarios.

- Recomendación de la siguiente mejor acción (recetas PERSONALIZED_ACTIONS)

  Para recomendar a sus usuarios la siguiente mejor acción en tiempo real, como suscribirse a su programa de fidelización o solicitar una tarjeta de crédito, debe utilizar una receta PERSONALIZED_ACTIONS. Al recomendar la siguiente mejor acción, puede aumentar la fidelidad de los clientes, generar más ingresos y mejorar la experiencia de los usuarios.

- Obtención de segmentos de usuarios (recetas USER_SEGMENTATION)

  Para obtener segmentos de usuarios en función de los datos de entrada de los elementos, como los usuarios que probablemente interactúan con elementos con un atributo determinado, debe utilizar una receta USER_SEGMENTATION. Obtener segmentos de usuarios puede ayudarle a crear campañas de marketing avanzadas que promocionen diferentes elementos entre distintos segmentos de usuarios en función de la probabilidad de que estos lleven a cabo alguna acción.

Recetas de Amazon Personalize

Amazon Personalize ofrece los siguientes tipos de recetas. Además de las diferencias de comportamiento, cada tipo tiene requisitos distintos para obtener recomendaciones, tal y como se muestra en la siguiente tabla.

Tipo de fórmula	Recetas	API	Requisitos de API
USER_PERSONALIZATION	User-Personalization-v2 User-Personalization Receta HRNN (heredada) Receta HRNN-Metadata (heredada) Receta HRNN-ColdStart (heredada)	GetRecommendations	`userId`: obligatorio `itemId`: no se utiliza `inputList` : N/D
POPULAR_ITEMS	Trending-Now Popularity-Count	GetRecommendations	`userId`: obligatorio solo si aplica un filtro que lo requiera `itemId`: no se utiliza `inputList` : N/D
PERSONALIZED_RANKING	Personalized-Ranking-v2 Personalized-Ranking	GetPersonalizedRanking	`userId`: obligatorio `itemId`: N/D `inputList` : lista de identificadores de elementos
RELATED_ITEMS	Similar-Items SIMS	GetRecommendations	`userId`: obligatorio solo si aplica

Tipo de fórmula	Recetas	API	Requisitos de API
	Receta de similitud semántica		un filtro que lo requiera `itemId`: obligatorio `inputList` : N/D
PERSONALIZED_ACTIONS	Next-Best-Action	GetActionRecommendations	`userId`: obligatorio `actionId`: no se utiliza `itemId`: no se utiliza `inputList` : N/D
SEGMENTACIÓN DE USUARIOS	Item-Affinity Item-Attribute-Affinity	CreateBatchSegmentJob	Para conocer los requisitos del flujo de trabajo por lotes, consulte Obtención de segmentos de usuario con un trabajo de segmentos por lotes.

Visualización de las recetas de Amazon Personalize disponibles

Para ver una lista de las recetas disponibles:

- En la consola de Amazon Personalize, elija un grupo de conjuntos de datos. En el panel de navegación, elija Soluciones y recetas y haga clic en la pestaña Recetas.
- Con el AWS SDK para Python (Boto3), llame a la ListRecipes API.
- Con el AWS CLI, usa el siguiente comando.

```
aws personalize list-recipes
```

Si desea obtener información sobre una receta mediante el SDK para Python (Boto3), llame a la API de DescribeRecipe. Para obtener información sobre una receta mediante el AWS CLI, utilice el siguiente comando.

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

Receta User-Personalization-v2

La receta User-Personalization-v2 (aws-user-personalization-v2) recomienda elementos con los que un usuario interactuará en función de las preferencias. Por ejemplo, puede usar User-Personalization-v2 para generar recomendaciones de películas personalizadas para una aplicación de streaming o recomendaciones de productos personalizadas para una aplicación de venta al por menor. Otros casos de uso incluyen la generación de recomendaciones en tiempo real para un sitio de noticias o recomendaciones por lotes para una campaña de marketing personalizada.

User-Personalization-v2 puede entrenar hasta cinco millones de elementos a partir de interacciones de elemento y conjuntos de datos de elementos. Y genera recomendaciones más relevantes con una latencia menor que User-Personalization.

Como User-Personalization-v2 recomienda los elementos más relevantes a los usuarios en función de los datos, recomienda con mayor frecuencia los elementos existentes con datos de interacciones. Para asegurarse de que las recomendaciones incluyen los elementos nuevos, puede utilizar una promoción que incluya algunos elementos en función de la marca temporal de creación. Para obtener más información sobre las promociones, consulte Promoción de elementos en las recomendaciones en tiempo real.

Esta receta utiliza una arquitectura basada en transformador para entrenar un modelo que aprenda el contexto y realice un seguimiento de las relaciones y los patrones de los datos. Los transformadores son un tipo de arquitectura de red neuronal que transforma o cambia una secuencia de entrada en una secuencia de salida. En el caso de Amazon Personalize, la secuencia de entrada es el historial de interacciones de elemento de un usuario en los datos. La secuencia de salida son las recomendaciones personalizadas. Para obtener más información sobre los transformadores, consulte ¿Qué son los transformadores en la inteligencia artificial? en el Centro de conceptos de computación en la nube de AWS.

User-Personalization-v2 utiliza un modelo de precios distinto al de otras recetas. Para obtener más información sobre el precio, consulte Precios de Amazon Personalize.

Temas

- Características de la receta
- Conjuntos de datos obligatorios y opcionales
- Propiedades e hiperparámetros

Características de la receta

User-Personalization-v2 utiliza las siguientes características de receta de Amazon Personalize al generar recomendaciones de elemento:

- Personalización en tiempo real: con este tipo de personalización, Amazon Personalize actualiza y adapta las recomendaciones de elemento en función de la evolución de los intereses del usuario. Para obtener más información, consulte Personalización en tiempo real.
- Exploración: con la exploración, las recomendaciones incluyen elementos con menos datos de interacciones o relevancia para el usuario. Con User-Personalization-v2, Amazon Personalize se encarga de la configuración de la exploración por usted. Para asegurarse de que las recomendaciones incluyen los elementos nuevos, puede utilizar promociones para incluir elementos nuevos en función de la marca temporal de creación. Para obtener más información sobre las promociones, consulte Promoción de elementos en las recomendaciones en tiempo real.
- Actualizaciones automáticas: con las actualizaciones automáticas, Amazon Personalize actualiza de forma automática el último modelo (versión de solución) cada dos horas para tener en cuenta nuevos elementos en las recomendaciones. Para obtener más información, consulte Actualizaciones automáticas.

- Metadatos con recomendaciones: con la receta User-Personalization-v2, si tiene un conjunto de datos de elementos con al menos una columna de metadatos, las campañas tienen automáticamente la opción de incluir metadatos de elemento con los resultados de las recomendaciones. No tiene que activar manualmente los metadatos de la campaña. Puede usar metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar los géneros de las películas a los carruseles. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.

Conjuntos de datos obligatorios y opcionales

Para usar User-Personalization-v2, debe crear un conjunto de datos de interacciones de elemento e importar al menos 1000 interacciones de elemento. Amazon Personalize genera recomendaciones basadas principalmente basadas en los datos de interacciones de elemento. Para obtener más información, consulte Datos de interacciones de elementos. User-Personalization-v2 puede entrenar hasta cinco millones de elementos entre interacciones de elemento y conjuntos de datos de elementos.

Con User-Personalization-v2, Amazon Personalize puede utilizar los datos de interacciones de elemento, entre los que se incluyen los siguientes:

- Tipo de evento y datos de valor de evento: Amazon Personalize utiliza los datos de tipo de evento, como los tipos de evento de clic y de visualización, para identificar la intención y el interés de los usuarios a través de cualquier patrón de comportamiento. Además, puede usar los datos de tipo y de valor de evento para filtrar los registros antes del entrenamiento. Para obtener más información, consulte Datos de valor de evento y tipo de evento.

  > **Note**
  >
  > Con User-Personalization-v2, el costo del entrenamiento se basa en los datos de interacciones antes de filtrarlos por tipo o valor de evento. Para obtener más información sobre el precio, consulte Precios de Amazon Personalize.

- Metadatos contextuales: son datos de interacciones que se recopilan en el entorno del usuario en el momento de un evento, como la ubicación o el tipo de dispositivo. Para obtener más información, consulte Metadatos contextuales.

Los siguientes conjuntos de datos son opcionales y pueden mejorar las recomendaciones:

- Conjunto de datos de usuarios: Amazon Personalize puede utilizar datos del conjunto de datos de usuarios para comprender mejor a los usuarios y los intereses que tienen. También puede usar los datos de un conjunto de datos de usuarios para filtrar las recomendaciones. Para obtener información sobre los datos de usuario que puede importar, consulte Metadatos de usuario.
- Conjunto de datos de elementos: Amazon Personalize puede utilizar los datos del conjunto de datos de elementos para identificar conexiones y patrones en el comportamiento. Esto ayuda a Amazon Personalize a entender a los usuarios y sus intereses. También puede usar los datos de un conjunto de datos de elementos para filtrar las recomendaciones. Para obtener información sobre los datos de elementos que puede importar, consulte Metadatos de elemento.

Propiedades e hiperparámetros

La receta User-Personalization-v2 tiene las siguientes propiedades:

- Nombre – `aws-user-personalization-v2`
- Nombre de recurso de Amazon (ARN) de la receta) – `arn:aws:personalize:::recipe/aws-user-personalization-v2`
- ARN del algoritmo – `arn:aws:personalize:::algorithm/aws-user-personalization-v2`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta User-Personalization-v2. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Con User-Personalization-v2, si activa el entrenamiento automático, Amazon Personalize realiza de forma automática la optimización de hiperparámetros cada 90 días. Sin un entrenamiento automático, no se produce dicha optimización.

La tabla proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)

Nombre	Descripción
Hiperparámetros de algoritmos	
`apply_recency_bias`	Determina si el modelo debe dar más ponderación a los datos de interacciones de elemento más recientes del conjunto de datos de interacciones de elemento. Los datos de las interacciones más recientes pueden incluir cambios repentinos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `apply_recency_bias` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `apply_recency_bias` en `false`. Valor predeterminado: #: `true` Rango: `true` o `false` Tipo de valor: booleano HPO ajustable: no

Receta User-Personalization

> **Important**
>
> Recomendamos utilizar la receta User-Personalization-v2. Puede tener en cuenta hasta cinco millones de elementos con un entrenamiento más rápido y generar recomendaciones más relevantes con una latencia más baja.

La receta de personalización del usuario (aws-user-personalization) está optimizada para todos los escenarios de recomendaciones personalizadas. Predice los elementos con los que es más probable que interactúe un usuario. Puede usar User-Personalization para generar recomendaciones

de películas personalizadas para una aplicación de streaming o recomendaciones de productos personalizadas para una aplicación de venta al por menor.

Con User-Personalization, Amazon Personalize genera recomendaciones basadas principalmente en los datos de interacciones de elemento de usuario de un conjunto de datos de interacciones de elemento. También puede usar metadatos de elemento y de usuario de los conjuntos de datos de elementos y usuarios. Para obtener más información sobre los datos que usa, consulte Conjuntos de datos obligatorios y opcionales.

Temas

- Características de la receta
- Conjuntos de datos obligatorios y opcionales
- Propiedades e hiperparámetros
- Entrenamiento con la receta User-Personalization (consola)
- Entrenamiento con la receta User-Personalization (SDK de Python)
- Obtención de recomendaciones y registro de impresiones (SDK para Python [Boto3])
- Cuaderno de Jupyter de ejemplo

Características de la receta

User-Personalization utiliza las siguientes características de receta de Amazon Personalize al generar recomendaciones de elemento:

- Personalización en tiempo real: con este tipo de personalización, Amazon Personalize actualiza y adapta las recomendaciones de elemento en función de la evolución de los intereses del usuario. Para obtener más información, consulte Personalización en tiempo real.
- Exploración: con la exploración, las recomendaciones incluyen elementos nuevos o elementos con menos datos de interacciones. Esto mejora la visibilidad de los elementos y su fidelización cuando se trata de un catálogo que cambia rápidamente o cuando los elementos nuevos, como elementos de noticias o promociones, son más relevantes para los usuarios cuando acaban de salir. Para obtener más información acerca de la exploración, consulte Exploration (Exploración).
- Actualizaciones automáticas: con las actualizaciones automáticas, Amazon Personalize actualiza de forma automática el último modelo (versión de solución) cada dos horas para tener en cuenta nuevos elementos en las recomendaciones. Para obtener más información, consulte Actualizaciones automáticas.

Conjuntos de datos obligatorios y opcionales

Para usar User-Personalization, debe crear un conjunto de datos de interacciones de elemento e importar al menos 1000 interacciones de elemento. Amazon Personalize genera recomendaciones basadas principalmente basadas en los datos de interacciones de elemento.

Con User-Personalization, Amazon Personalize puede utilizar los datos de interacciones de elemento, entre los que se incluyen los siguientes:

- Tipo de evento y datos de valor de evento: Amazon Personalize utiliza los datos de tipo de evento, como los tipos de evento de clic y de visualización, para identificar la intención y el interés de los usuarios a través de cualquier patrón de comportamiento. Además, puede usar los datos de tipo y de valor de evento para filtrar los registros antes del entrenamiento. Para obtener más información, consulte Datos de valor de evento y tipo de evento.
- Metadatos contextuales: son datos de interacciones que se recopilan en el entorno del usuario en el momento de un evento, como la ubicación o el tipo de dispositivo. Para obtener más información, consulte Metadatos contextuales.
- Datos de impresiones: las impresiones son listas de elementos que estaban visibles para un usuario cuando interactuaba con un elemento concreto (al hacer clic, verlo, comprarlo, etc.). Para obtener más información, consulte Datos de impresiones.

Los siguientes conjuntos de datos son opcionales y pueden mejorar las recomendaciones:

- Conjunto de datos de usuarios: Amazon Personalize puede utilizar datos del conjunto de datos de usuarios para comprender mejor a los usuarios y los intereses que tienen. También puede usar los datos de un conjunto de datos de usuarios para filtrar las recomendaciones. Para obtener información sobre los datos de usuario que puede importar, consulte Metadatos de usuario.
- Conjunto de datos de elementos: Amazon Personalize puede utilizar los datos del conjunto de datos de elementos para identificar conexiones y patrones en el comportamiento. Esto ayuda a Amazon Personalize a entender a los usuarios y sus intereses. También puede usar los datos de un conjunto de datos de elementos para filtrar las recomendaciones. Para obtener información sobre los datos de elementos que puede importar, consulte Metadatos de elemento.

Propiedades e hiperparámetros

La receta User-Personalization incluye las siguientes propiedades:

- Nombre: `aws-user-personalization`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-user-personalization`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-user-personalization`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta User-Personalization. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en HPO?

Name	Description (Descripción)
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones de elementos incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Para decidir el valor óptimo, utilice HPO. Para utilizar HPO,

Name	Description (Descripción)
	establezca `performHPO` en `true` cuando llame a las operaciones CreateSolution y CreateSolutionVersion. Valor predeterminado: 149 Rango: [32, 256] Tipo de valor: entero HPO ajustable: sí
`bptt`	Determina si se debe utilizar la propagación hacia atrás a través de la técnica de tiempo. La propagación hacia atrás a través del tiempo es una técnica que actualiza ponderaciones en algoritmos basados en redes neuronales recurrentes. Utilice `bptt` para créditos a largo plazo para conectar recompensas retrasadas a eventos tempranos. Por ejemplo, una recompensa retrasada puede ser una compra realizada después de varios clics. Un evento temprano puede ser un clic inicial. Incluso dentro de los mismos tipos de eventos, tales como un clic, es buena idea tener en cuenta los efectos a largo plazo y maximizar las recompensas totales. Para tener en cuenta los efectos a largo plazo, utilice valores mayores de `bptt`. El uso de un valor de `bptt` mayor requiere conjuntos de datos más grandes y más tiempo de procesamiento. Valor predeterminado: 32 Rango: [2, 32] Tipo de valor: entero HPO ajustable: sí

Name	Description (Descripción)
`recency_mask`	Determina si el modelo debe tener en cuenta las últimas tendencias de popularidad en el conjunto de datos de interacciones de elementos. Las últimas tendencias de popularidad pueden incluir cambios bruscos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `recency_mask` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `recency_mask` en `false`. Para obtener buenas recomendaciones con el mismo peso, es posible que necesite un conjunto de datos de entrenamiento mayor. Valor predeterminado: `True` Rango: `True` o `False` Tipo de valor: booleano HPO ajustable: sí
Hiperparámetros de caracterización	

Name	Description (Descripción)
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min_user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Name	Description (Descripción)
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas, ya que los datos de estos usuarios tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min_user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,99 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no
Hiperparámetros de configuración de la campaña de exploración de elementos	

Name	Description (Descripción)
`exploration_weight`	Determina la frecuencia con la que las recomendaciones incluyen elementos con menos datos de interacciones de elementos o relevancia. Cuanto más se acerque el valor a 1,0, mayor será la exploración. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia). Para obtener más información, consulte the section called “CampaignConfig”. Valor predeterminado: 0,3 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Name	Description (Descripción)
`exploration_item_age_cut_of f`	Especifique la antigüedad máxima del elemento en días desde la última interacción en todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad de un elemento en función de su marca temporal de creación o, si faltan estos datos, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad de un elemento, consulte Datos de la marca de tiempo de creación. Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó. Valor predeterminado: 30,0 Rango: flotantes positivos Tipo de valor: flotante HPO ajustable: no

Entrenamiento con la receta User-Personalization (consola)

Para usar la receta User-Personalization para generar recomendaciones en la consola, primero entrene una nueva versión de la solución con la receta. A continuación, implemente una campaña con la versión de la solución y utilícela para obtener recomendaciones.

Entrenamiento de una versión de la solución con la receta User-Personalization (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. Cree un grupo de conjuntos de datos personalizado con un esquema nuevo y cargue su conjunto de datos con datos de impresiones. Si lo desea, incluya CREATION_TIMESTAMP y datos Metadatos de texto no estructurado en su conjunto de datos de elementos para que Amazon Personalize pueda calcular con mayor precisión la antigüedad de un elemento e identificar los elementos fríos.

   Para obtener más información sobre la importación de datos, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.
3. En la página Grupos de conjuntos de datos, elija el nuevo grupo de conjuntos de datos que contenga el conjunto de datos o los conjuntos de datos con datos de impresiones.
4. En el panel de navegación, seleccione Soluciones y recetas y elija Crear solución.
5. En la página Crear solución, en Nombre de la solución, introduzca el nombre de la nueva solución.
6. En Tipo de solución, elija Recomendación de elementos para obtener recomendaciones de elementos para sus usuarios.
7. En Receta, elija aws-user-personalization. Aparece la sección Configuración de la solución con varias opciones de configuración.
8. En Configuración adicional, si el conjunto de datos de interacciones con artículos tiene la columna EVENT_TYPE o tanto EVENT_TYPE como EVENT_VALUE, si lo desea, utilice los campos Tipo de evento y Umbral del valor del evento para elegir los datos de interacciones con artículos que Amazon Personalize usará al entrenar el modelo. Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento.

   Si tienes varios tipos de eventos y utilizas User-Personalization-v 2 o Personalized-Ranking-v 2 recetas, también puedes especificar distintos pesos para los distintos tipos. Por ejemplo, puede configurar una solución para dar más peso a los eventos de compra que a los de clic. Para obtener más información, consulte Optimización de una solución con la configuración de eventos.
9. Si lo desea, configure los hiperparámetros para su solución. Para obtener una lista de las propiedades e hiperparámetros de la receta User-Personalization, consulte Propiedades e hiperparámetros.
10. Elija Crear y entrenar la solución para empezar el entrenamiento. Aparece la página Panel.

Puede acceder a la página de detalles de la solución para realizar un seguimiento del progreso del entrenamiento en la sección Versiones de la solución. Una vez finalizado el entrenamiento, el estado es Activo.

Creación de una campaña y obtención de recomendaciones (consola)

Cuando el estado de la versión de la solución sea Activo, estará listo para crear su campaña y recibir recomendaciones del modo siguiente:

1. En la página de detalles de la solución o en la página Campañas, seleccione Crear nueva campaña.
2. En la página Crear nueva campaña, en los Detalles de la campaña, proporcione la siguiente información:
   - Nombre de la campaña: escriba el nombre de la campaña. El texto que escriba aquí aparecerá en el panel Campaña y en la página de detalles.
   - Solución: elija la solución que acaba de crear.
   - ID de versión de la solución: elija el ID de la versión de la solución que acaba de crear.
   - Transacciones aprovisionadas mínimas por segundo: defina las transacciones aprovisionadas mínimas por segundo que admite Amazon Personalize. Para obtener más información, consulte la operación CreateCampaign.
3. Para Configuración de recursos, proporcione la siguiente información:
   - Ponderación de la exploración: configure cuánto explorar, donde las recomendaciones incluyen elementos con menos datos de interacciones o relevancia con más frecuencia cuanta más exploración especifique. Cuanto más se acerque el valor a 1, mayor será la exploración. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).
   - Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elementos, en días desde la última interacción, para definir el alcance de la exploración del elemento. Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior.

     Por ejemplo, si especifica 10, durante la exploración solo se tendrán en cuenta los elementos con datos de interacciones de elementos de los 10 días transcurridos desde la última interacción del conjunto de datos.

> **Note**
>
> Las recomendaciones pueden incluir elementos sin datos de interacciones de elementos de fuera de este período de tiempo. El motivo es que estos elementos son relevantes para los intereses del usuario y no se requirió a la exploración que los identificara.

4. Elija Crear una campaña.
5. En la página de detalles de la campaña, cuando el estado de la campaña es Activo, puede utilizarla para obtener recomendaciones y registrar las impresiones. Para obtener más información, consulte Paso 5: Obtener recomendaciones en “Introducción”.

   Amazon Personalize actualiza automáticamente la última versión de la solución cada dos horas para incluir nuevos datos. La campaña utiliza automáticamente la versión actualizada de la solución. Para obtener más información, consulte Actualizaciones automáticas.

   Para actualizar la campaña manualmente, primero debe crear y entrenar una nueva versión de la solución mediante la consola o la operación CreateSolutionVersion, con la configuración `trainingMode` establecida en `update`. A continuación, actualice la campaña manualmente en la página Campaña de la consola o mediante la operación UpdateCampaign.

   > **Note**
   >
   > Amazon Personalize no actualiza automáticamente las versiones de soluciones que haya creado antes del 17 de noviembre de 2020.

Entrenamiento con la receta User-Personalization (SDK de Python)

Cuando haya creado un grupo de conjuntos de datos y cargado sus conjuntos de datos con datos de impresiones, podrá entrenar una solución con la receta User-Personalization. Si lo desea, incluya CREATION_TIMESTAMP y datos Metadatos de texto no estructurado en su conjunto de datos de elementos para que Amazon Personalize pueda calcular con mayor precisión la antigüedad de un elemento e identificar los elementos fríos. Para obtener más información sobre la creación de grupos de conjuntos de datos y la carga de datos de entrenamiento, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Para entrenar una solución con la receta de personalización del usuario mediante el SDK AWS

1. Cree una nueva solución mediante el método `create_solution`.

   Sustituya `solution name` por el nombre de la solución y `dataset group arn` por el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos.

   ```
   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)
   ```

   Para obtener una lista de las propiedades e hiperparámetros de la aws-user-personalization receta, consulte. Propiedades e hiperparámetros

2. Cree una nueva versión de la solución con los datos de entrenamiento actualizados y configure `trainingMode` en `FULL` para usar el siguiente fragmento de código. Sustituya `solution arn` por el ARN de la solución.

   ```
   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. Cuando Amazon Personalize termine de crear la versión de la solución, cree la campaña con los siguientes parámetros:

- Proporcione un nuevo `campaign name` y el `solution version arn` generado en el paso 2.
- Modifique el hiperparámetro de configuración de exploración de elementos `explorationWeight` para configurar cuánto explorar. Los elementos con menos datos de interacciones de elementos o relevancia se recomiendan con más frecuencia cuanto más se acerca el valor a 1,0. El valor predeterminado es 0,3.
- Modifique el parámetro de hiperparámetro de la configuración de exploración de elementos `explorationItemAgeCutOff` para proporcionar la duración máxima, en días en relación con la última interacción, durante la cual se deben explorar los elementos. Cuanto mayor sea el valor, más elementos se tendrán en cuenta durante la exploración.

Use el siguiente fragmento de código de Python para crear una nueva campaña con énfasis en la exploración con un límite de exploración de 30 días. La creación de una campaña suele hacerse en unos minutos, pero puede tardar más de una hora.

```
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)
```

Con User-Personalization, Amazon Personalize actualiza automáticamente la versión de la solución cada dos horas para incluir nuevos datos. La campaña utiliza automáticamente la versión actualizada de la solución. Para obtener más información, consulte Actualizaciones automáticas.

Para actualizar la campaña manualmente, primero debe crear y entrenar una nueva versión de la solución mediante la consola o la operación CreateSolutionVersion, con la configuración

`trainingMode` establecida en `update`. A continuación, actualice la campaña manualmente en la página Campaña de la consola o mediante la operación UpdateCampaign.

> **Note**
>
> Amazon Personalize no actualiza automáticamente las versiones de soluciones que haya creado antes del 17 de noviembre de 2020.

Obtención de recomendaciones y registro de impresiones (SDK para Python [Boto3])

Una vez creada la campaña, puede usarla para obtener recomendaciones para un usuario y registrar las impresiones. Para obtener información sobre cómo obtener recomendaciones de lotes, AWS SDKs consulteCreación de un trabajo de inferencia por lotes (AWS SDKs).

Para obtener recomendaciones y registrar las impresiones

1. Llame al método `get_recommendations`. Cambie `campaign arn` al ARN de la nueva campaña y `user id` al identificador de usuario del usuario.

```
import boto3

rec_response = personalize_runtime.get_recommendations(campaignArn = 'campaign
 arn', userId = 'user id')
print(rec_response['recommendationId'])
```

2. Cree un nuevo rastreador de eventos para enviar PutEvents solicitudes. Reemplace `event tracker name` por el nombre de su rastreador de eventos y `dataset group arn` por el ARN de su grupo de conjuntos de datos.

```
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. Use `recommendationId` del paso 1 y `event tracking id` del paso 2 para crear una nueva solicitud `PutEvents`. Esta solicitud registra los nuevos datos de impresión de la sesión del usuario. Cambie `user id` por el identificador del usuario.

```
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']],
     }]
)
```

Cuaderno de Jupyter de ejemplo

Para ver una muestra de cuaderno de Jupyter que describe el uso de la receta User-Personalization, consulte Personalización de usuario con la exploración.

Receta Trending-Now

La receta Trending-Now (aws-trending-now) genera recomendaciones para artículos que se están volviendo cada vez más populares entre los usuarios. Puede utilizar la receta Trending-Now si los elementos que están ganando popularidad son más relevantes para sus clientes. Por ejemplo, es posible que sus clientes valoren en gran medida aquello con que interactúan otros usuarios. Entre los usos habituales se incluyen la recomendación contenido viral en las redes sociales, elementos de noticias de última hora o vídeos deportivos recientes.

Trending-Now identifica automáticamente los elementos que están ganando popularidad calculando el aumento de las interacciones de cada uno de ellos a lo largo de intervalos de tiempo configurables. Los elementos con la tasa de aumento más alta se consideran elementos populares.

El tiempo se basa en los datos de marca temporal del conjunto de datos de interacciones de elementos. Los elementos considerados provienen de los datos de interacciones que se han importado de forma masiva e incremental. No es necesario crear manualmente una nueva versión de la solución para que Trending-Now tenga en cuenta nuevos elementos en los datos de interacciones.

Puede especificar el intervalo de tiempo proporcionando `Trend discovery frequency` al crear la solución. Por ejemplo, si especifica `30 minutes` para `Trend discovery frequency` por cada 30 minutos de datos, Amazon Personalize identifica los elementos con la mayor tasa de aumento de interacciones desde la última evaluación. Las posibles frecuencias incluyen 30 minutos, 1 hora, 3 horas y 1 día. Elija una frecuencia que se ajuste a la distribución de los datos de sus interacciones. La falta de datos durante el intervalo que elija puede reducir la precisión de las recomendaciones. Si importa cero interacciones en los dos últimos intervalos de tiempo, Amazon Personalize recomienda solo los elementos populares en lugar de los elementos que estén ganando popularidad.

Con Trending-Now, puede llamar a la operación GetRecommendations u obtener recomendaciones en la página de Campaña de prueba de la consola Amazon Personalize. Amazon Personalize devuelve los elementos que más popularidad están ganando. Solo se aprueba una solicitud `userId` si se aplica un filtro que lo requiera. Con la GetRecommendations API, puede configurar el número de elementos populares devueltos con el parámetro. `numResults` No puede obtener recomendaciones por lotes con la receta Trending-Now.

Para usar Trending-Now, debe crear un conjunto de datos de interacciones entre elementos con al menos 1000 interacciones históricas y de eventos únicas combinadas (después de filtrar por EventType y eventValueThreshold, si se proporciona). Al generar recomendaciones de elementos de tendencia, Trending-Now no utiliza los datos de los conjuntos de datos de elementos o usuarios. Sin embargo, aún puede filtrar las recomendaciones en función de los datos de estos conjuntos de datos. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Temas

- Propiedades e hiperparámetros
- Creación de una solución (SDK para Python (Boto3))
- Cuaderno de Jupyter de ejemplo

Propiedades e hiperparámetros

La receta Trending-Now incluye las siguientes propiedades:

- Nombre: `aws-trending-now`

- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-trending-now`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-trending-now-custom`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta Trending-Now. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en HPO?

Name	Description (Descripción)
Hiperparámetros de transformación de características	
`Trend discovery frequency`	Especifique la frecuencia con la que Amazon Personalize evalúa los datos de sus interacciones e identifica los elementos de tendencia. Por ejemplo, si especifica `30 minutes` para `Trend discovery frequency` , Amazon Personalize identifica cada 30 minutos los elementos con la mayor tasa de aumento de interacciones en intervalos de 30 minutos. Las frecuencias disponibles incluyen 30 minutos, 1 hora, 2 horas, 3 horas y 1 día. Elija una frecuencia que se ajuste a la distribución de los datos de sus interacciones. La falta de datos durante el intervalo que elija puede reducir la precisión de las recomendaciones. Si utiliza la operación de CreateSolution API y

Name	Description (Descripción)
	no especifica ningún valor, el valor predeterminado es cada 2 horas. **Note** Para cualquier valor de parámetro superior a 2 horas, Amazon Personalize actualiza automáticamente las recomendaciones de artículos más populares cada 2 horas para tener en cuenta las nuevas interacciones y los nuevos artículos. Valor predeterminado: 2 horas Valores posibles: 30 minutos, 1 hora, 2 horas, 3 horas y 1 día. Tipo de valor: cadena HPO ajustable: no

Creación de una solución (SDK para Python (Boto3))

Los siguientes códigos muestran cómo crear una solución con la receta Trending-Now mediante el SDK for Python (Boto3). Los valores posibles de `trend_discovery_frequency` son `30 minutes`, `1 hour`, `3 hours` y `1 day`. Para obtener información sobre cómo crear una tabla manualmente con la consola, consulte Creación de una solución (consola).

```
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'])
```

Cuaderno de Jupyter de ejemplo

Para ver un ejemplo de bloc de notas de Jupyter que muestre cómo usar la receta Trending-Now, consulte trending_now_example.ipynb en el repositorio de muestras de Amazon Personalize. GitHub

Receta Popularity-Count

Popularity-Count recomienda los elementos más populares en función de los datos de sus interacciones. Los elementos más populares son aquellos con más datos de interacciones de usuarios únicos. La receta devuelve los mismos elementos populares para todos los usuarios. Popularity-Count es una buena base de referencia para la comparación con otras recetas utilizando las métricas de evaluación que Amazon Personalize genera al crear una versión de la solución. Para obtener más información, consulte Evaluación de una versión de solución de Amazon Personalize con métricas.

Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con Popularity-Count, debe crear manualmente una nueva versión de solución (volver a entrenar el modelo) para que Amazon Personalize tenga en cuenta los nuevos elementos para las recomendaciones y actualice el modelo con el comportamiento más reciente de los usuarios. A continuación, debe actualizar cualquier campaña con la versión de la solución. Para obtener más información, consulte Mantenimiento de la relevancia de las recomendaciones.

Esta receta predefinida tiene las siguientes propiedades:

- Nombre – `aws-popularity-count`
- ARN de receta: `arn:aws:personalize:::recipe/aws-popularity-count`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-popularity-count`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/sims`
- Tipo de receta: `USER_PERSONALIZATION`

Popularity-Count no expone ningún hiperparámetro.

Receta Personalized-Ranking-v2

La receta Personalized-Ranking-v2 genera clasificaciones personalizadas de los elementos. Una clasificación personalizada es una lista de elementos recomendados que se vuelven a clasificar por relevancia para un usuario específico. Resulta útil si tiene una colección de elementos pedidos, como resultados de búsqueda, promociones o listas seleccionadas, y quiere ofrecer una nueva clasificación personalizada para cada uno de sus usuarios.

Personalized-Ranking-v2 puede entrenar hasta cinco millones de elementos a partir de interacciones de elemento y conjuntos de datos de elementos. Y genera clasificaciones más precisas con una latencia menor que Personalized-Ranking.

Al utilizar Personalized-Ranking-v2, especifique los elementos que se van a clasificar en una operación de la API GetPersonalizedRanking. Si especifica elementos sin datos de interacciones, Amazon Personalize devolverá estos elementos sin una puntuación de recomendación en la respuesta de la API GetPersonalizedRanking.

Esta receta utiliza una arquitectura basada en transformador para entrenar un modelo que aprenda el contexto y realice un seguimiento de las relaciones y los patrones de los datos. Los transformadores son un tipo de arquitectura de red neuronal que transforma o cambia una secuencia de entrada en una secuencia de salida. En el caso de Amazon Personalize, la secuencia de entrada es el historial de interacciones de elemento de un usuario en los datos. La secuencia de salida son las recomendaciones personalizadas. Para obtener más información sobre los transformadores, consulte What Are Transformers In Artificial Intelligence? en el Centro de conceptos de computación en la nube de AWS.

Personalized-Ranking-v2 utiliza un modelo de precios distinto al de otras recetas. Para obtener más información sobre el precio, consulte Precios de Amazon Personalize.

Temas

- Características de la receta
- Conjuntos de datos obligatorios y opcionales
- Propiedades e hiperparámetros

Características de la receta

Personalized Ranking-v2 utiliza las siguientes características de receta de Amazon Personalize al clasificar los elementos:

- Personalización en tiempo real: con este tipo de personalización, Amazon Personalize actualiza y adapta las recomendaciones de elemento en función de la evolución de los intereses del usuario. Para obtener más información, consulte Personalización en tiempo real.
- Metadatos con recomendaciones: con la receta Personalized-Ranking-v2, si tiene un conjunto de datos de elementos con al menos una columna de metadatos, las campañas tienen automáticamente la opción de incluir metadatos de elemento con los resultados de las recomendaciones. No tiene que activar manualmente los metadatos de la campaña. Puede usar metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar los géneros de las películas a los carruseles. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.

Conjuntos de datos obligatorios y opcionales

Para usar Personalized-Ranking-v2, debe crear un conjunto de datos de interacciones de elemento e importar al menos 1000 interacciones de elemento. Amazon Personalize genera clasificaciones basadas principalmente en los datos de interacciones de elemento. Para obtener más información, consulte Datos de interacciones de elementos. Personalized-Ranking-v2 puede entrenar hasta cinco millones de elementos entre interacciones de elemento y conjuntos de datos de elementos.

Con Personalized-Ranking-v2, Amazon Personalize puede utilizar los datos de interacciones de elemento, entre los que se incluyen los siguientes:

- Tipo de evento y datos de valor de evento: Amazon Personalize utiliza los datos de tipo de evento, como los tipos de evento de clic y de visualización, para identificar la intención y el interés de los usuarios a través de cualquier patrón de comportamiento. Además, puede usar los datos de tipo y de valor de evento para filtrar los registros antes del entrenamiento. Para obtener más información, consulte Datos de valor de evento y tipo de evento.

> **Note**
>
> Con Personalized-Ranking-v2, el costo del entrenamiento se basa en los datos de interacciones antes de filtrarlos por tipo o valor de evento. Para obtener más información sobre el precio, consulte Precios de Amazon Personalize.

- Metadatos contextuales: son datos de interacciones que se recopilan en el entorno del usuario en el momento de un evento, como la ubicación o el tipo de dispositivo. Para obtener más información, consulte Metadatos contextuales.

Los siguientes conjuntos de datos son opcionales y pueden mejorar las recomendaciones:

- Conjunto de datos de usuarios: Amazon Personalize puede utilizar datos del conjunto de datos de usuarios para comprender mejor a los usuarios y los intereses que tienen. También puede usar los datos de un conjunto de datos de usuarios para filtrar las recomendaciones. Para obtener información sobre los datos de usuario que puede importar, consulte Metadatos de usuario.
- Conjunto de datos de elementos: Amazon Personalize puede utilizar los datos del conjunto de datos de elementos para identificar conexiones y patrones en el comportamiento. Esto ayuda a Amazon Personalize a entender a los usuarios y sus intereses. También puede usar los datos de un conjunto de datos de elementos para filtrar las recomendaciones. Para obtener información sobre los datos de elementos que puede importar, consulte Metadatos de elemento.

Propiedades e hiperparámetros

La receta Personalized-Ranking-v2 tiene las siguientes propiedades:

- Nombre – `aws-personalized-ranking-v2`
- Nombre de recurso de Amazon (ARN) de la receta) – `arn:aws:personalize:::recipe/aws-personalized-ranking-v2`
- ARN del algoritmo – `arn:aws:personalize:::algorithm/aws-personalized-ranking-v2`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta Personalized-Ranking-v2. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento

del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Con Personalized-Ranking-v2, si activa el entrenamiento automático, Amazon Personalize realiza de forma automática la optimización de hiperparámetros cada 90 días. Sin un entrenamiento automático, no se produce dicha optimización.

La tabla proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)

Nombre	Descripción
Hiperparámetros de algoritmos	
`apply_recency_bias`	Determina si el modelo debe dar más ponderación a los datos de interacciones de elemento más recientes del conjunto de datos de interacciones de elemento. Los datos de las interacciones más recientes pueden incluir cambios repentinos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `apply_recency_bias` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `apply_recency_bias` en `false`. Valor predeterminado: #: `true` Rango: `true` o `false` Tipo de valor: booleano HPO ajustable: no

Receta Personalized-Ranking

> **Important**
>
> Recomendamos utilizar la receta Personalized-Ranking-v2. Puede tener en cuenta hasta cinco millones de elementos con un entrenamiento más rápido y generar clasificaciones más precisas con una latencia más baja.

La receta Personalized-Ranking genera clasificaciones personalizadas de elementos. Una clasificación personalizada es una lista de elementos recomendados que se vuelven a clasificar para un usuario específico. Resulta útil si tiene una colección de elementos pedidos, como resultados de búsqueda, promociones o listas seleccionadas, y quiere ofrecer una nueva clasificación personalizada para cada uno de sus usuarios. Por ejemplo, con Personalized-Ranking, Amazon Personalize puede volver a clasificar los resultados de búsqueda que genere con OpenSearch.

Para entrenar un modelo, la receta Personalized-Ranking utiliza los datos de su conjunto de datos de interacciones de elementos y, si los creó, los conjuntos de datos de elementos y usuarios de su grupo de conjuntos de datos (estos conjuntos de datos son opcionales). Con Personalized-Ranking, su conjunto de datos de elementos puede incluir Metadatos de texto no estructurado y su conjunto de datos de interacciones puede incluir Metadatos contextuales. Para obtener clasificaciones personalizadas, use la API GetPersonalizedRanking.

Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con Personalized-Ranking, debe crear manualmente una nueva versión de solución (reentrenar el modelo) para que Amazon Personalize tenga en cuenta los nuevos elementos para las recomendaciones y actualizar el modelo con el comportamiento más reciente del usuario. A continuación, debe actualizar cualquier campaña con la versión de la solución. Para obtener más información, consulte Mantenimiento de la relevancia de las recomendaciones.

> **Note**
>
> Si proporciona elementos sin datos de interacciones para su clasificación, Amazon Personalize devolverá estos elementos sin una puntuación de recomendación en la respuesta de la API GetPersonalizedRanking.

Esta receta tiene las siguientes propiedades:

- Nombre – `aws-personalized-ranking`
- Nombre de recurso de Amazon (ARN) de la receta) – `arn:aws:personalize:::recipe/aws-personalized-ranking`
- ARN del algoritmo – `arn:aws:personalize:::algorithm/aws-personalized-ranking`
- ARN de transformación de características – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Tipo de receta – `PERSONALIZED_RANKING`

Hiperparámetros

En la siguiente tabla se describen los hiperparámetros de la receta Personalize-Ranking. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en optimización hiperparámetros (HPO)?

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones de elementos incluya patrones más complicados. El uso de más

Nombre	Descripción
	dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Para decidir el valor óptimo, utilice HPO. Para utilizar HPO, establezca `performHPO` en `true` cuando llame a las operaciones CreateSolution y CreateSolutionVersion. Valor predeterminado: 149 Rango: [32, 256] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`bptt`	Determina si se debe utilizar la propagación hacia atrás a través de la técnica de tiempo. La propagaci ón hacia atrás a través del tiempo es una técnica que actualiza ponderaciones en algoritmos basados en redes neuronales recurrentes. Utilice `bptt` para créditos a largo plazo para conectar recompens as retrasadas a eventos tempranos. Por ejemplo, una recompensa retrasada puede ser una compra realizada después de varios clics. Un evento temprano puede ser un clic inicial. Incluso dentro de los mismos tipos de eventos, tales como un clic, es buena idea tener en cuenta los efectos a largo plazo y maximizar las recompensas totales. Para tener en cuenta los efectos a largo plazo, utilice valores mayores de `bptt`. El uso de un valor de `bptt` mayor requiere conjuntos de datos más grandes y más tiempo de procesami ento. Valor predeterminado: 32 Rango: [2, 32] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`recency_mask`	Determina si el modelo debe tener en cuenta las últimas tendencias de popularidad en el conjunto de datos de interacciones de elementos. Las últimas tendencias de popularidad pueden incluir cambios bruscos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `recency_mask` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `recency_mask` en `false`. Para obtener buenas recomendaciones con el mismo peso, es posible que necesite un conjunto de datos de entrenamiento mayor. Valor predeterminado: #: `True` Rango: `True` o `False` Tipo de valor: booleano HPO ajustable: sí
Hiperparámetros de caracterización	

Nombre	Descripción
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Nombre	Descripción
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas, ya que los datos de estos usuarios tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,99 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Cuaderno de ejemplo de Personalized-Ranking

Para ver una muestra de cuaderno de Jupyter que describe el uso de la receta Personalized-Ranking, consulte el ejemplo de Personalize Ranking.

Receta de similitud semántica

La receta de similitud semántica (aws-semantic-similarity) genera recomendaciones para elementos que son semánticamente similares a un elemento determinado en función del contenido textual. A diferencia de las recetas de similitud tradicionales que se basan en las interacciones entre el usuario y el elemento, esta receta analiza las descripciones textuales y los atributos de los elementos para generar incrustaciones e identificar elementos semánticamente similares

Esta receta es ideal para situaciones en las que quieres recomendar elementos en función de la similitud del contenido, como por ejemplo, recomendar libros con temas similares, artículos sobre temas relacionados o productos con descripciones similares. Funciona especialmente bien para artículos nuevos con un historial de interacciones limitado (escenarios de inicio en frío) y para catálogos en los que las relaciones semánticas son más importantes que los patrones de coocurrencia.

Con Semantic-Silarity, usted proporciona un identificador de artículo en una GetRecommendationsoperación (o en la consola Amazon Personalize) y Amazon Personalize devuelve una lista de artículos similares. O bien, puedes usar un flujo de trabajo por lotes para obtener artículos similares para todos los artículos de tu inventario (consulta Cómo obtener recomendaciones de artículos por lotes).

Temas

- Características de la receta
- Conjuntos de datos obligatorios y opcionales
- Propiedades e hiperparámetros

Características de la receta

Semantic-Silarity utiliza las siguientes funciones de recetas de Amazon Personalize al generar recomendaciones de artículos:

- Personalización en tiempo real: con la receta de similitud semántica, Amazon Personalize mantiene tu catálogo de artículos actualizado automáticamente. Cuando agrega nuevos artículos a su conjunto de datos de artículos o actualiza los metadatos de los artículos existentes, estos cambios se reflejan en sus recomendaciones en aproximadamente 30 minutos si utiliza el entrenamiento gradual. Esto garantiza que tus clientes siempre vean los artículos más recientes disponibles en tu catálogo sin necesidad de intervención manual ni de esperar a que se complete un ciclo de reentrenamiento. Esto resulta especialmente útil para los catálogos que cambian

con frecuencia, como artículos de noticias, publicaciones de blogs o ofertas de productos de temporada. Para habilitar las actualizaciones incrementales, los clientes deben:

- `performIncrementalUpdate`Configúrelo `true` como para la solución en la API
- Elija la opción «Entrenamiento completo e incremental» o «Entrenamiento incremental» en Método de entrenamiento en la interfaz de usuario

Tenga en cuenta que habilitar las actualizaciones incrementales generará costos adicionales cada vez que se realice una actualización.

- Metadatos con recomendaciones: con la fórmula de similitud semántica, las campañas tienen automáticamente la opción de incluir los metadatos de los artículos en los resultados de las recomendaciones. No tiene que activar manualmente los metadatos de la campaña. Puede usar metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar los géneros de las películas a los carruseles. Para obtener más información, consulta los metadatos de los elementos en las recomendaciones.

Conjuntos de datos obligatorios y opcionales

Para usar la receta de similitud semántica, debes crear un conjunto de datos de elementos. Amazon Personalize genera recomendaciones basadas en el significado semántico de los metadatos de los artículos. Para obtener más información, consulte Metadatos del artículo. La similitud semántica puede procesar hasta 10 millones de elementos de tu conjunto de datos de elementos.

Con Semantic-Silarity, Amazon Personalize requiere datos de artículos que incluyan lo siguiente

- Campo ItemName: un campo de cadena `itemName` establecido en. `true` Este campo debe contener el título o el nombre del elemento.
- Campo de descripción textual: al menos un campo de cadena marcado como `textual` que contiene la descripción del elemento. Debe ser el campo que mejor describa y represente el elemento.

Amazon Personalize usa este campo para generar incrustaciones semánticas que capturen el significado y el contenido de sus artículos.

Además, debe configurar el campo reservado CREATION_TIMESTAMP si desea utilizar una clasificación basada en la frescura. Para obtener más información, consulte Propiedades e hiperparámetros.

Los siguientes conjuntos de datos son opcionales y pueden mejorar las recomendaciones:

- Conjunto de datos de interacciones: Amazon Personalize puede usar los datos de su conjunto de datos de interacciones para calcular las puntuaciones de popularidad en función de la interacción de los usuarios con los artículos. Puede utilizar las puntuaciones de popularidad para clasificar artículos similares en función de su popularidad entre los usuarios. Debes proporcionar un conjunto de datos de interacciones si quieres utilizar una clasificación basada en la popularidad. También puedes usar los datos de un conjunto de datos de interacciones para filtrar las recomendaciones. Para obtener información sobre los datos de interacción que puede importar, consulte Datos de interacción entre elementos

Propiedades e hiperparámetros

La receta de similitud semántica tiene las siguientes propiedades:

- Nombre: `aws-semantic-similarity`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-semantic-similarity`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-semantic-similarity`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/aws-semantic-similarity`
- Tipo de receta: `RELATED_ITEMS`

Para obtener más información, consulte Elección de una receta.

La receta de similitud semántica no tiene hiperparámetros expuestos, pero puedes configurar los factores de popularidad y frescura al crear una campaña para influir en la clasificación de artículos similares.

La tabla proporciona la siguiente información para cada factor:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)

Name	Description (Descripción)
Antigüedad	El factor de frescura representa qué tan reciente es un artículo. La frescura se calcula normalizando la antigüedad del artículo en función de su CREATION_TIMESTAMP. Para usar el factor de frescura, debes incluir el campo CREATION_TIMESTAMP en el esquema del conjunto de datos de Items. Los valores más altos del factor de frescura darán prioridad a los elementos más nuevos entre las recomendaciones semánticamente similares Valor predeterminado: `0.0` Rango: `[0.0, 1.0]` Tipo de valor: doble
Popularidad	El factor de popularidad representa la popularidad de un artículo en función de las interacciones de los usuarios. La popularidad se calcula normalizando el número de interacciones que recibe cada artículo. Para usar el factor de popularidad, debes incluir un conjunto de datos de interacciones al crear tu grupo de conjuntos de datos. Los valores más altos del factor de popularidad dan prioridad a los artículos con más interacciones con los clientes entre recomendaciones semánticamente similares. Valor predeterminado: `0.0` Rango: `[0.0, 1.0]` Tipo de valor: doble

Ten en cuenta que las puntuaciones de frescura y popularidad se calculan durante el entrenamiento y que las actualizaciones incrementales no actualizan las puntuaciones de popularidad y frescura. Para que los factores de popularidad y frescura más recientes influyan en la clasificación de los

artículos recomendados, utilice el reentrenamiento automático o el reentrenamiento manual de la solución y actualice la campaña con la nueva versión de la solución.

Receta Similar-Items

> **Note**
>
> Todas las recetas RELATED_ITEMS utilizan datos de interacciones. Elija Similar-Items si también tiene metadatos de elementos y quiere que Amazon Personalize los use para buscar elementos similares. O bien, elija Receta SIMS si desea configurar más hiperparámetros para el modelo.

La receta Similar-Items (aws-similar-items) genera recomendaciones para artículos que son similares a un artículo que especifique. Use Similar-Items para ayudar a los clientes a descubrir nuevos elementos en el catálogo en función de su comportamiento anterior y de los metadatos de los elementos. La recomendación de elementos similares puede aumentar la participación de los usuarios, la tasa de clics y la tasa de conversión de su aplicación.

Similar-Items calcula la similitud en función de los datos de interacciones y de cualquier metadato de elemento que proporcione. Tiene en cuenta la coexistencia del elemento en los historiales de los usuarios de su conjunto de datos de interacciones y cualquier similitud en los metadatos del elemento. Por ejemplo, con Similar-Items, Amazon Personalize podría recomendar elementos que los clientes compren con frecuencia juntos con un estilo similar (Metadatos categóricos) o películas que diferentes usuarios también hayan visto con una descripción similar (Metadatos de texto no estructurado).

Con Similar-Items, proporciona un identificador de elemento en una operación GetRecommendations (o en la consola Amazon Personalize) y Amazon Personalize devuelve una lista de elementos similares. También puede usar un flujo de trabajo por lotes para obtener elementos similares para todos los elementos de tu inventario (consulte Obtención de recomendaciones de elemento por lotes). Cuando obtenga elementos similares, puede filtrarlos en función de un atributo del elemento que especifique en la solicitud. Para ello, añada un elemento `CurrentItem.attribute` al filtro. Para ver un ejemplo, consulta item data filter examples.

Para usar Similar-Items, debe crear un conjunto de datos de interacciones de elementos con al menos 1000 interacciones históricas y de eventos únicas (combinadas). Para obtener predicciones más precisas, le recomendamos que también cree un conjunto de datos de elementos e importe

metadatos sobre los elementos de su catálogo. Similar-Items no utiliza los datos de un conjunto de datos de usuarios al generar recomendaciones. Aún puede filtrar las recomendaciones en función de los datos de un conjunto de datos de usuarios. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Si tiene un conjunto de datos de elementos con datos textuales y datos de títulos de elementos, puede generar temas para elementos relacionados en las recomendaciones por lotes. Para obtener más información, consulte Recomendaciones por lotes con temas del Generador de contenidos

Puede obtener recomendaciones de elementos similares a un elemento inactivo (un elemento con menos de cinco interacciones). Si Amazon Personalize no encuentra el identificador del elemento que especificó en la solicitud de recomendación o en el archivo de entrada del lote, la receta devolverá los elementos populares como recomendaciones.

Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con Similar-Items, debe crear manualmente una nueva versión de solución (reentrenar el modelo) para que Amazon Personalize tenga en cuenta los nuevos elementos para las recomendaciones y actualizar el modelo con el comportamiento más reciente del usuario. A continuación, debe actualizar cualquier campaña con la versión de la solución. Para obtener más información, consulte Mantenimiento de la relevancia de las recomendaciones.

Propiedades e hiperparámetros

La receta Similar-Items incluye las siguientes propiedades:

- Nombre: `aws-similar-items`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-similar-items`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-similar-items`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta Similar-Items. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en HPO?

Name	Description (Descripción)
Hiperparámetros de algoritmos	
`popularity_discount_factor`	Configure cómo la popularidad influye en las recomendaciones. Especifique un valor cercano a cero para incluir los elementos más populares. Especifique un valor cercano a uno para hacer menos hincapié en la popularidad. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no
`item_id_hidden_dim`	El número de variables ocultas que Amazon Personalize utiliza para modelar las inserciones de identificadores de elementos en función de los datos de interacciones. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Para usar `item_id_hidden_dim` , debe usar HPO y proporcionar valores de rango mínimos y máximos. Amazon Personalize utiliza HPO para encontrar el mejor valor dentro del rango que especifique. Especifique un valor máximo mayor cuando tenga un conjunto de datos de interacciones de elementos grande. El uso de un valor máximo mayor requiere más tiempo de procesamiento.

Name	Description (Descripción)
	Para utilizar HPO, establezca `performHPO` en `true` cuando llame a la operación CreateSolution. Valor predeterminado: 100 Rango: [30, 200] Tipo de valor: entero HPO ajustable: sí
`item_metadata_hidden_dim`	El número de variables ocultas que Amazon Personalize utiliza para modelar los metadatos de los elementos. Para usar `item_metadata_hidden_dim`, debe usar HPO y proporcionar valores de rango mínimos y máximos. Amazon Personalize utiliza HPO para encontrar el mejor valor dentro del rango que especifique. Especifique un valor máximo mayor cuando tenga un conjunto de datos de interacciones de elementos grande. El uso de un valor máximo mayor requiere más tiempo de procesamiento. Para utilizar HPO, establezca `performHPO` en `true` cuando llame a la operación CreateSolution. Valor predeterminado: 100 Rango: [30, 200] Tipo de valor: entero HPO ajustable: sí

Receta SIMS

> **Note**
>
> Todas las recetas RELATED_ITEMS utilizan datos de interacciones. Elija SIMS si desea configurar más hiperparámetros para el modelo. Elija Receta Similar-Items si tiene metadatos de elementos y quiere que Amazon Personalize los use para buscar elementos similares.

La receta de Item-to-item similitudes (SIMS) utiliza el filtrado colaborativo para recomendar los elementos que se parecen más a los elementos que especificas al recibir recomendaciones. SIMS utiliza su conjunto de datos de interacciones de elementos, no los metadatos de los elementos, como el color o el precio, para determinar la similitud. SIMS identifica la presencia simultánea del elemento en los historiales de los usuarios de su conjunto de datos de interacciones para recomendar elementos similares. Por ejemplo, con SIMS, Amazon Personalize podría recomendar elementos de cafetería que los clientes compren juntos con frecuencia o películas que otros usuarios también hayan visto.

Cuando obtenga recomendaciones de elementos similares, puede filtrarlos en función de un atributo del elemento que especifique en la solicitud. Para ello, añada un elemento `CurrentItem.attribute` al filtro. Para ver un ejemplo, consulta item data filter examples.

Para usar SIMS, debe crear un conjunto de datos de interacciones de elementos con al menos 1000 interacciones históricas y de eventos únicas (combinadas). SIMS no utiliza los datos de un conjunto de datos de usuarios o elementos al generar recomendaciones. Aún puede filtrar las recomendaciones en función de los datos de estos conjuntos de datos. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Si no hay suficientes datos sobre el comportamiento de los usuarios para un elemento o no se encuentra el identificador del elemento que ha proporcionado, SIMS le recomendará los elementos más populares. Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con SIMS, debe crear manualmente una nueva versión de solución (reentrenar el modelo) para que Amazon Personalize tenga en cuenta los nuevos elementos para las recomendaciones y actualizar el modelo con el comportamiento más reciente del usuario. A continuación, debe actualizar cualquier campaña con la versión de la solución. Para obtener más información, consulte Mantenimiento de la relevancia de las recomendaciones.

La receta SIMS incluye las siguientes propiedades:

- Nombre: `aws-sims`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-sims`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-sims`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/sims`
- Tipo de receta: `RELATED_ITEMS`

En la siguiente tabla se describen los hiperparámetros de la receta SIMS. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en optimización hiperparámetros (HPO)?

Name	Description (Descripción)
Hiperparámetros de algoritmos	
`popularity_discount_factor`	Configure cómo la popularidad influye en las recomendaciones. Especifique un valor cercano a cero para incluir los elementos más populares. Especifique un valor cercano a uno para hacer menos hincapié en la popularidad. Valor predeterminado: 0,5 Rango: [0,0, 1,0] Tipo de valor: flotante

Name	Description (Descripción)
	HPO ajustable: sí
`min_cointeraction_count`	El número mínimo de cointeracciones que necesita para calcular la similitud entre un par de elementos. Por ejemplo, un valor de 3 significa que necesita tres o más usuarios que interactuaron con ambos elementos para que el algoritmo calcule su similitud. Valor predeterminado: 3 Rango: [0, 10]. Tipo de valor: entero HPO ajustable: sí
Hiperparámetros de caracterización	

Name	Description (Descripción)
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos disponibles de un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Valor predeterminado: 0,005 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Name	Description (Descripción)
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos disponibles en un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas. Los usuarios con un historial largo tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, `min_hist_length_percentile = 0.05` y `max_hist_length_percentile = 0.95` incluyen todos los usuarios excepto aquellos cuyas longitudes de historial están en el 5 % inferior o superior. Valor predeterminado: 0,995 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Name	Description (Descripción)
`min_item_interaction_count_percentile`	El percentil mínimo de interacción del elemento cuenta para incluirlo en el modelo de entrenamiento. Utilice `min_item_interaction_count_percentile` para excluir un porcentaje de elementos con un historial breve de interacciones. Los elementos con un historial breve suelen ser elementos nuevos. Eliminarlos permite entrenar modelos que se centren más en elementos con un historial conocido. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los elementos, pero que elimine los casos de borde. Valor predeterminado: 0,01 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Name	Description (Descripción)
`max_item_interaction_count_percentile`	El percentil máximo de interacción del elemento cuenta para incluirlo en el modelo de entrenamiento. Utilice `max_item_interaction_count_percentile` para excluir un porcentaje de elementos con un largo historial de interacciones. Los elementos con un historial largo tienden a ser más antiguos y podrían estar obsoletos. Por ejemplo, una película que está agotada. La eliminación de estos elementos permite centrarse en elementos más relevantes. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos configurar un valor que retenga la mayoría de los elementos, pero elimine los casos de borde. Por ejemplo, `min_item_interaction_count_percentile = 0.05` y `max_item_interaction_count_percentile = 0.95` incluyen todos los elementos excepto aquellos con un recuento de interacción que esté en el 5 % inferior o superior. Valor predeterminado: 0,9 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Cuaderno de ejemplo de SIMS

Para ver un cuaderno de Jupyter de muestra que describe cómo utilizar la receta SIMS, consulte Búsqueda de elementos similares + HPO.

Receta Next-Best-Action

La receta Next-Best-Action (aws-next-best-action) genera recomendaciones en tiempo real sobre las siguientes mejores acciones para los usuarios. La siguiente mejor acción para un usuario es aquella que probablemente realice. Por ejemplo, inscribirse en su programa de fidelización, descargar su aplicación o solicitar una tarjeta de crédito.

Con Next-Best-Action, puede recomendar acciones personalizadas a los usuarios cuando utilicen su aplicación. Sugerir la acción correcta para un usuario puede hacer que un número mayor de estos realicen sus acciones. En función de las acciones que quiera recomendar, puede incrementar la fidelidad de los clientes, generar más ingresos y mejorar la experiencia del usuario con su aplicación. Para ver un ejemplo de caso de uso que describe cómo las recomendaciones de acciones personalizadas pueden beneficiar a una aplicación de comercio electrónico, consulte Ejemplo de casos de uso.

Amazon Personalize predice la siguiente mejor acción a partir de las acciones que importa en su conjunto de datos de acciones. Identifica las acciones que es más probable que realice un usuario en función de sus interacciones con las acciones y los elementos. Si los datos de la acción incluyen el valor de esta, Amazon Personalize tiene en cuenta el valor de la acción. Si un usuario tiene la misma probabilidad de realizar dos acciones distintas, Amazon Personalize clasifica la acción con el valor más alto.

Cuando recibe recomendaciones de acciones en tiempo real para un usuario, Amazon Personalize devuelve una lista de las acciones que es más probable que realice el usuario dentro de un período de tiempo configurable (llamado `action optimization period`). Por ejemplo, las acciones que es más probable que realice en los próximos 14 días. La lista está ordenada en orden descendente por puntuación de propensión. Esta puntuación representa la probabilidad de que el usuario realice la acción.

Hasta que se importen los datos de interacción de acción, Amazon Personalize recomienda acciones sin personalización y las puntuaciones de propensión son de 0,0. Una acción tendrá una puntuación cuando tenga lo siguiente:

- Al menos 50 interacciones de acción con el tipo de evento TAKEN.
- Al menos 50 interacciones de acción con el tipo de evento NOT_TAKEN o VIEWED.

Estas interacciones de acción deben estar presentes en el entrenamiento de la versión de solución más reciente y deben producirse en un intervalo de seis semanas a partir de la última marca temporal de interacción del conjunto de datos de interacciones de acción.

Para obtener más información sobre los datos que utiliza la receta Next-Best-Action, consulte Conjuntos de datos obligatorios y opcionales.

Al crear una solución con la receta Next-Best-Action, puede configurar el intervalo de tiempo que Amazon Personalize utiliza al predecir las acciones mediante el hiperparámetro de caracterización `action optimization period`. Para obtener más información, consulte Propiedades e hiperparámetros.

Temas

- Ejemplo de casos de uso
- Características de la receta
- Conjuntos de datos obligatorios y opcionales
- Propiedades e hiperparámetros

Ejemplo de casos de uso

Sugerir la acción correcta para un usuario puede hacer que un número mayor de estos realicen sus acciones. En función de las acciones que quiera recomendar, posiblemente pueda incrementar la fidelidad de los clientes, generar más ingresos y mejorar la experiencia del usuario con su aplicación.

Por ejemplo, puede que tenga una aplicación de comercio electrónico que sugiera las siguientes acciones:

- Suscribirse al programa de fidelización
- Descargar la aplicación móvil
- Comprar en la categoría Joyas
- Comprar en la categoría Belleza y cuidado

Puede que tenga un usuario que compre con frecuencia en su sitio y que haya realizado repetidamente acciones de compra de Joyas y Belleza y cuidado. Para este usuario, las recomendaciones de acción de Amazon Personalize y sus puntuaciones pueden incluir lo siguiente:

- Suscribirse al programa de fidelización

Puntuación de propensión: 1,00

- Comprar en la categoría Joyas

  Puntuación de propensión: 0,86

- Comprar en la categoría Belleza y cuidado

  Puntuación de propensión: 0,85

Con estas recomendaciones de acción, sabe pedirle al usuario que se inscriba en su programa de fidelización. Esta acción tiene la puntuación de propensión más alta y es aquella que es más probable que realice el usuario. Esto se debe a que el usuario compra en su tienda con frecuencia y probablemente aproveche las ventajas del programa de fidelización.

Características de la receta

La receta Next-Best-Action usa las siguientes características de receta de Amazon Personalize al generar recomendaciones de acciones:

- Personalización en tiempo real: Amazon Personalize utiliza la personalización en tiempo real para actualizar y adaptar las recomendaciones de acciones en función de la evolución de los intereses de los usuarios. Para obtener más información, consulte Personalización en tiempo real.
- Exploración: con la exploración, las recomendaciones incluyen nuevas acciones o bien acciones con menos datos de interacciones. Para obtener más información acerca de la exploración, consulte Exploration (Exploración).
- Actualizaciones automáticas: con las actualizaciones automáticas, Amazon Personalize actualiza de forma automática el último modelo (versión de la solución) cada dos horas para incluir nuevas acciones en las recomendaciones a través de la exploración. Para obtener más información, consulte Actualizaciones automáticas.

Conjuntos de datos obligatorios y opcionales

Para usar la receta Next-Best-Action, debe crear los siguientes conjuntos de datos:

- Acciones: importa datos sobre sus acciones, como su valor, en un conjunto de datos de acciones de Amazon Personalize.

En los datos de las acciones, puede proporcionar un valor EXPIRATION_TIMESTAMP para cada acción. Si una acción ha caducado, Amazon Personalize no la incluirá en las recomendaciones. También puede proporcionar un valor REPEAT_FREQUENCY para cada una de ellas. Esto indica cuánto tiempo debe esperar Amazon Personalize antes de volver a recomendar una acción después de que un usuario interactúe con ella. Para obtener información sobre los datos que puede almacenar un conjunto de datos de acciones, consulte Metadatos de acción.

- Interacciones de elementos: su conjunto de datos de interacciones de elementos debe tener un mínimo de 1000 interacciones de elementos. Amazon Personalize utiliza las interacciones de los elementos para comprender el estado actual de los usuarios y sus intereses. Para obtener información sobre los datos de interacciones de los elementos, consulte Datos de interacciones de elementos.

Los siguientes conjuntos de datos son opcionales:

- Conjunto de datos de interacciones de acción: una interacción de acción es una interacción en la que participan un usuario y una acción del conjunto de datos de acciones. Puede importar las interacciones de acción con estado Realizado, No realizado y Visto. Aunque estos datos son opcionales, se recomienda importar los datos de interacción de acciones para obtener recomendaciones de calidad. Si no tiene datos de interacción de acciones, puede crear un conjunto de datos de interacciones de este tipo vacío y registrar las interacciones de los clientes con las acciones mediante la operación de la API PutActionInteractions.

  Hasta que se importen los datos de interacción de acción, Amazon Personalize recomienda acciones sin personalización y las puntuaciones de propensión son de 0,0. Una acción tendrá una puntuación cuando tenga lo siguiente:

  - Al menos 50 interacciones de acción con el tipo de evento TAKEN.
  - Al menos 50 interacciones de acción con el tipo de evento NOT_TAKEN o VIEWED.

  Estas interacciones de acción deben estar presentes en el entrenamiento de la versión de solución más reciente y deben producirse en un intervalo de seis semanas a partir de la última marca temporal de interacción del conjunto de datos de interacciones de acción.

  Para obtener información sobre los datos de interacciones de acción que puede importar, consulte Datos de interacciones de acción. Para obtener información sobre cómo registrar eventos de interacción de acciones, consulte Registro de eventos de interacción de acciones en tiempo real.

> **Note**
>
> Con Next-Best-Action, Amazon Personalize no utiliza datos de impresiones ni metadatos contextuales en un conjunto de datos de interacciones de acción.

- Usuarios: Amazon Personalize utiliza cualquier dato del conjunto de datos de sus usuarios para comprender mejor a los usuarios y sus intereses. También puede usar los datos de un conjunto de datos de usuarios para filtrar las recomendaciones de acción. Para obtener información sobre los datos de usuario que puede importar, consulte Metadatos de usuario.
- Elementos: Amazon Personalize utiliza los datos del conjunto de datos de elementos junto con el conjunto de datos de interacciones de elementos para identificar conexiones y patrones en su comportamiento. Esto ayuda a Amazon Personalize a entender a los usuarios y sus intereses. Para obtener información sobre los datos de elementos que puede importar, consulte Metadatos de elemento.

Propiedades e hiperparámetros

La receta Next-Best-Action no admite la optimización de hiperparámetros. La receta Next-Best-Action tiene las siguientes propiedades:

- Nombre – `aws-next-best-action`
- Nombre de recurso de Amazon (ARN) de la receta) – `arn:aws:personalize:::recipe/aws-next-best-action`
- ARN del algoritmo – `arn:aws:personalize:::algorithm/aws-next-best-action`

En la siguiente tabla se describen los hiperparámetros de caracterización de la receta aws-next-best-action. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: indica si el parámetro puede participar en HPO.

Nombre	Descripción
Hiperparámetros de caracterización	
`action_optimization_period`	El período de tiempo que Amazon Personalize utiliza al predecir las siguientes mejores acciones para un usuario. Por ejemplo, las acciones que es más probable que el usuario realice en los próximos 14 días. Si no dispone de muchos datos de interacción de acciones, especifique un valor más grande. Si no sabe con seguridad qué valor debe especificar, utilice el valor predeterminado. Valor predeterminado: 14 Rango: [7, 28] Tipo de valor: entero HPO ajustable: no

Receta Item-Affinity

La receta Item-Affinity (aws-item-affinity) es una receta USER_SEGMENTATION que crea un segmento de usuarios (grupo de usuarios) para cada elemento que especifique. Estos son los usuarios que Amazon Personalize predice que probablemente interactuarán con cada elemento. Use Item-Affinity para obtener más información sobre sus usuarios y tomar medidas en función de sus respectivos segmentos de usuarios.

Por ejemplo, es posible que desee crear una campaña de marketing para su aplicación de venta al por menor basada en las preferencias de los usuarios respecto a los elementos de su catálogo. Item-Affinity crearía un segmento de usuarios para cada elemento en función de los datos de sus conjuntos de datos de interacciones y elementos. Puede recurrir a ello para promocionar diferentes elementos entre distintos segmentos de usuarios en función de la probabilidad de que realicen una acción (por ejemplo, hacer clic en un elemento o comprarlo). Otros usos pueden incluir la venta

cruzada de productos a diferentes grupos de usuarios o la identificación de posibles candidatos para un puesto de trabajo.

Para obtener segmentos de usuarios basados en elementos, debe crear una solución y una versión de la solución con la receta Item-Affinity, agregar una lista de elementos en formato JSON a un bucket de Amazon S3 y crear un trabajo de segmento por lotes. Amazon Personalize envía un segmento de usuarios para cada elemento a su ubicación de salida en Amazon S3. Los datos de entrada pueden tener un máximo de 500 elementos para obtener segmentos de usuarios. Para obtener información sobre cómo preparar los datos de entrada para un trabajo de segmento por lotes, consulte Preparación de los datos de entrada para las recomendaciones por lotes.

Debe tener un conjunto de datos de interacciones de elementos para usar Item-Affinity. Los conjuntos de datos de elementos y usuarios son opcionales. Puede obtener segmentos de usuarios con trabajos de segmentos por lotes. Para obtener más información, consulte Obtención de segmentos de usuarios por lotes.

Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con Item-Affinity, debe crear una nueva versión de solución para que Amazon Personalize tenga en cuenta nuevos usuarios para los segmentos de usuarios y actualizar el modelo con el comportamiento más reciente de los usuarios. Para obtener un segmento de usuarios para un elemento, este debe haber estado presente en el momento de crear la versión de la solución.

La receta Item-Affinity incluye las siguientes propiedades:

- Nombre – `aws-item-affinity`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-item-affinity`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-item-affinity`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/item-affinity`
- Tipo de receta: `USER_SEGMENTATION`

En la siguiente tabla se describen los hiperparámetros de la receta Item-Affinity. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. No puede usar la optimización de hiperparámetros (HPO) con la receta Item-Affinity.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Valor predeterminado: 149 Rango: [32, 256] Tipo de valor: entero

Receta Item-Attribute-Affinity

La receta Item-Attribute-Affinity (aws-item-attribute-affinity) es una receta USER_SEGMENTATION que crea un segmento de usuarios (grupo de usuarios) para cada atributo de elemento que especifique. Estos son los usuarios que Amazon Personalize predice que es más probable que interactúen con elementos con el atributo en particular. Use Item-Attribute-Affinity para obtener más información sobre sus usuarios y tomar medidas en función de sus respectivos segmentos de usuarios.

Por ejemplo, es posible que desee crear una campaña de marketing para su aplicación de venta al por menor basada en las preferencias de los usuarios respecto a los tipos de zapato de su catálogo. Item-Attribute-Affinity crearía un segmento de usuarios para cada tipo de zapato en función de los datos de sus conjuntos de datos de interacciones y elementos. Puede recurrir a ello para promocionar diferentes zapatos entre distintos segmentos de usuarios en función de la probabilidad

de que realicen una acción (por ejemplo, hacer clic en un zapato o comprarlo). Otros usos pueden incluir la promoción de diferentes géneros cinematográficos entre distintos usuarios o la identificación de posibles candidatos a un puesto de trabajo en función del tipo de trabajo.

Para obtener segmentos de usuarios basados en los atributos de los elementos, debe crear una solución y una versión de la solución con la receta Item-Attribute-Affinity, agregar una lista de elementos en formato JSON a un bucket de Amazon S3 y crear un trabajo de segmento por lotes. Amazon Personalize envía un segmento de usuarios para cada elemento a su ubicación de salida en Amazon S3. Los datos de entrada pueden tener un máximo de 10 consultas, cada una de las cuales consiste en uno o más atributos de un elemento. Para obtener información sobre cómo preparar los datos de entrada para un trabajo de segmento por lotes, consulte Preparación de los datos de entrada para las recomendaciones por lotes.

Debe tener un conjunto de datos de interacciones de elementos y otro de elementos para usar Item-Attribute-Affinity. Su conjunto de datos de elementos debe tener al menos una columna que sea una columna de metadatos no textual ni reservada. Puede obtener segmentos de usuarios con trabajos de segmentos por lotes. Para obtener más información, consulte Obtención de segmentos de usuarios por lotes con recursos personalizados.

Después de crear una versión de la solución, asegúrese de mantener la versión y los datos de la solución actualizados. Con Item-Attribute-Affinity, debe crear una nueva versión de solución para que Amazon Personalize tenga en cuenta nuevos usuarios para los segmentos de usuarios y actualizar el modelo con el comportamiento más reciente de los usuarios. Para obtener un segmento de usuarios para el atributo de un elemento, este debe haber estado presente en el momento de crear la versión de la solución.

La receta Item-Attribute-Affinity incluye las siguientes propiedades:

- Nombre – `aws-item-attribute-affinity`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-item-attribute-affinity`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-item-attribute-affinity`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/item-attribute-affinity`
- Tipo de receta: `USER_SEGMENTATION`

En la siguiente tabla se describen los hiperparámetros de la receta Item-Attribute-Affinity. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. No puede usar la optimización de hiperparámetros (HPO) con la receta Item-Attribute-Affinity.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Valor predeterminado: 149 Rango: [32, 256] Tipo de valor: entero

Recetas de HRNN heredadas

Las recetas heredadas de HRNN ya no están disponibles. Esta documentación sirve como referencia.

Recomendamos usar la receta aws-user-personalizaton (personalizada por el usuario) en lugar de las recetas antiguas de HRNN. User-Personalization mejora y unifica la funcionalidad que ofrecen las recetas HRNN. Para obtener más información, consulte Receta User-Personalization.

Amazon Personalize puede seleccionar automáticamente la receta de red neuronal recurrente jerárquica (HRNN) más adecuada en función de su análisis de los datos de entrada. Esta opción se denomina AutoML. Para obtener más información, consulte Uso de AutoML para elegir una receta HRNN (solo API).

Temas

- Uso de AutoML para elegir una receta HRNN (solo API)
- Receta HRNN (heredada)
- Receta HRNN-Metadata (heredada)
- Receta HRNN-ColdStart (heredada)

Uso de AutoML para elegir una receta HRNN (solo API)

Amazon Personalize puede seleccionar automáticamente la receta de red neuronal recurrente jerárquica (HRNN) más adecuada en función de su análisis de los datos de entrada. Esta opción se denomina AutoML. Para realizar AutoML, establezca el parámetro `performAutoML` en `true` cuando llame a la API CreateSolution.

También puede especificar la lista de recetas que Amazon Personalize examina para determinar la receta óptima, en función de una métrica que especifique. En este caso, debe llamar a la operación `CreateSolution`, especificar `true` en el parámetro `performAutoML`, omitir el parámetro `recipeArn` e incluir el parámetro `solutionConfig`, especificando `metricName` y `recipeList` como parte del objeto `autoMLConfig`.

En la tabla siguiente, se muestra cómo se elige una receta. Es necesario especificar `performAutoML` o `recipeArn`, pero no ambos. AutoML solo se realiza mediante las recetas HRNN.

performAutoML	recipeArn	solutionConfig	Resultado
true	omitido	omitido	Amazon Personalize elige la receta
true	omitido	`autoMLConfig` : `metricName` y `recipeList` especificados	Amazon Personalize elige una receta de la lista que optimiza la métrica

performAutoML	recipeArn	solutionConfig	Resultado
omitido	especificado	omitido	Usted especifica la receta.
omitido	especificado	especificado	Usted especifica la receta y anula las propiedades de entrenamiento predeterminadas.

> **Note**
>
> Cuando `performAutoML` es `true`, se omiten todos los parámetros del objeto `solutionConfig` excepto `autoMLConfig`.

Receta HRNN (heredada)

> **Note**
>
> Las recetas heredadas de HRNN ya no están disponibles. Esta documentación sirve como referencia.
> Recomendamos que utilice la receta aws-user-personalizaton (User-Personalization) en lugar de las recetas heredadas de HRNN. User-Personalization mejora y unifica la funcionalidad que ofrecen las recetas HRNN. Para obtener más información, consulte Receta User-Personalization.

La receta de la red neuronal recurrente jerárquica (HRNN) de Amazon Personalize modela los cambios en el comportamiento de los usuarios para proporcionar recomendaciones durante una sesión. Una sesión es un conjunto de interacciones de usuarios durante un periodo de tiempo determinado con el objetivo de encontrar un elemento específico para satisfacer una necesidad, por ejemplo. Al ponderar con mayor puntuación las interacciones recientes de un usuario, puede proporcionar recomendaciones más relevantes durante una sesión.

HRNN se adapta a la intención y los intereses de los usuarios, que pueden cambiar con el tiempo. Toma historias de usuario ordenadas y las pondera automáticamente para realizar mejores

deducciones. HRNN utiliza un mecanismo de regulación para modelar las ponderaciones de descuentos como una función aprendible de los elementos y la marca de tiempo.

Amazon Personalize deriva las características de cada usuario de su conjunto de datos. Si se ha realizado la integración de datos en tiempo real, estas características se actualizan en tiempo real en función de la actividad de los usuarios. Para obtener una recomendación, proporcione solo el `USER_ID`. Si también proporciona `ITEM_ID`, Amazon Personalize lo ignorará.

La receta HRNN incluye las siguientes propiedades:

- Nombre – `aws-hrnn`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-hrnn`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-hrnn`
- ARN de transformación de características – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Tipo de receta: `USER_PERSONALIZATION`

En la siguiente tabla se describen los hiperparámetros de la receta HRNN. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en HPO?

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras

Nombre	Descripción
	de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones de elementos incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Para decidir el valor óptimo, utilice HPO. Para utilizar HPO, establezca `performHPO` en `true` cuando llame a las operaciones CreateSolution y CreateSolutionVersion. Valor predeterminado: 43 Rango: [32, 256] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`bptt`	Determina si se debe utilizar la propagación hacia atrás a través de la técnica de tiempo. La propagación hacia atrás a través del tiempo es una técnica que actualiza ponderaciones en algoritmos basados en redes neuronales recurrentes. Utilice `bptt` para créditos a largo plazo para conectar recompensas retrasadas a eventos tempranos. Por ejemplo, una recompensa retrasada puede ser una compra realizada después de varios clics. Un evento temprano puede ser un clic inicial. Incluso dentro de los mismos tipos de eventos, tales como un clic, es buena idea tener en cuenta los efectos a largo plazo y maximizar las recompensas totales. Para tener en cuenta los efectos a largo plazo, utilice valores mayores de `bptt`. El uso de un valor de `bptt` mayor requiere conjuntos de datos más grandes y más tiempo de procesamiento. Valor predeterminado: 32 Rango: [2, 32] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`recency_mask`	Determina si el modelo debe tener en cuenta las últimas tendencias de popularidad en el conjunto de datos de interacciones de elementos. Las últimas tendencias de popularidad pueden incluir cambios bruscos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `recency_mask` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `recency_mask` en `false`. Para obtener buenas recomendaciones con el mismo peso, es posible que necesite un conjunto de datos de entrenamiento mayor. Valor predeterminado: #: `True` Rango: `True` o `False` Tipo de valor: booleano HPO ajustable: sí
Hiperparámetros de caracterización	

Nombre	Descripción
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Nombre	Descripción
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas, ya que los datos de estos usuarios tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,99 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Receta HRNN-Metadata (heredada)

> **Note**
>
> Las recetas heredadas de HRNN ya no están disponibles. Esta documentación sirve como referencia.

> Recomendamos que utilice la receta aws-user-personalizaton (User-Personalization) en lugar de las recetas heredadas de HRNN. User-Personalization mejora y unifica la funcionalidad que ofrecen las recetas HRNN. Para obtener más información, consulte Receta User-Personalization.

La receta HRNN-Metadata predice los elementos con los que interactuará un usuario. Es igual que la receta HRNN, pero tiene otras características derivadas de metadatos contextuales, de usuario y de elemento (conjuntos de datos interacciones, usuarios y elementos, respectivamente). HRNN-Metadata ofrece mayor precisión que los modelos sin metadatos cuando se dispone de metadatos de alta calidad. El uso de esta receta puede requerir tiempos de entrenamiento más largos.

La receta HRNN-Metadata incluye las siguientes propiedades:

- Nombre – `aws-hrnn-metadata`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-hrnn-metadata`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-hrnn-metadata`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/featurize_metadata`
- Tipo de receta: `USER_PERSONALIZATION`

En la siguiente tabla se describen los hiperparámetros de la receta HRNN-Metadata. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en optimización hiperparámetros (HPO)?

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones de elementos incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Para decidir el valor óptimo, utilice HPO. Para utilizar HPO, establezca `performHPO` en `true` cuando llame a las operaciones CreateSolution y CreateSolutionVersion. Valor predeterminado: 43 Rango: [32, 256] Tipo de valor: entero HPO ajustable: sí
`bptt`	Determina si se debe utilizar la propagación hacia atrás a través de la técnica de tiempo. La propagaci ón hacia atrás a través del tiempo es una técnica que actualiza ponderaciones en algoritmos basados en redes neuronales recurrentes. Utilice `bptt` para créditos a largo plazo para conectar recompens as retrasadas a eventos tempranos. Por ejemplo, una recompensa retrasada puede ser una compra realizada después de varios clics. Un evento temprano puede ser un clic inicial. Incluso dentro de los mismos tipos de eventos, tales como un clic, es buena idea tener en cuenta los efectos a largo plazo y maximizar las recompensas totales. Para tener en cuenta los

Nombre	Descripción
	efectos a largo plazo, utilice valores mayores de `bptt`. El uso de un valor de `bptt` mayor requiere conjuntos de datos más grandes y más tiempo de procesami ento. Valor predeterminado: 32 Rango: [2, 32] Tipo de valor: entero HPO ajustable: sí
`recency_mask`	Determina si el modelo debe tener en cuenta las últimas tendencias de popularidad en el conjunto de datos de interacciones de elementos. Las últimas tendencias de popularidad pueden incluir cambios bruscos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `recency_mask` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriore s, establezca `recency_mask` en `false`. Para obtener buenas recomendaciones con el mismo peso, es posible que necesite un conjunto de datos de entrenamiento mayor. Valor predeterminado: #: `True` Rango: `True` o `False` Tipo de valor: booleano HPO ajustable: sí
Hiperparámetros de caracterización	

Nombre	Descripción
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Nombre	Descripción
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas, ya que los datos de estos usuarios tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,99 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Receta HRNN-ColdStart (heredada)

> **Note**
>
> Las recetas heredadas de HRNN ya no están disponibles. Esta documentación sirve como referencia.

> Recomendamos que utilice la receta aws-user-personalizaton (User-Personalization) en lugar de las recetas heredadas de HRNN. User-Personalization mejora y unifica la funcionalidad que ofrecen las recetas HRNN. Para obtener más información, consulte Receta User-Personalization.

Utilice la receta HRNN-Coldstart para predecir los elementos con los que interactuará un usuario cuando añada nuevos elementos e interacciones con frecuencia y desee recibir recomendaciones para esos elementos de inmediato. La receta HRNN-Coldstart es similar a la receta HRNN-Metadata, pero le permite obtener recomendaciones de nuevos elementos.

Además, puede utilizar la receta HRNN-Coldstart cuando desee excluir del entrenamiento los elementos que tengan una larga lista de interacciones, ya sea porque se trate de una tendencia de popularidad reciente o porque las interacciones podrían ser muy inusuales e introducir ruido en el entrenamiento. Con HRNN-Coldstart, puede filtrar los elementos menos relevantes y crear un subconjunto para el entrenamiento. El subconjunto de elementos, que se denominan elementos fríos, son elementos que tienen eventos de interacción relacionados en el conjunto de datos de interacciones de elementos. Un elemento se considera un elemento frío cuando tiene:

- Menos interacciones que un número máximo de interacciones especificado. Especifique este valor en el hiperparámetro `cold_start_max_interactions` de la receta.
- Una duración relativa más corta que la duración máxima. Especifique este valor en el hiperparámetro `cold_start_max_duration` de la receta.

Para reducir el número de elementos fríos, establezca un valor inferior para `cold_start_max_interactions` o `cold_start_max_duration`. Para aumentar el número de elementos fríos, establezca un valor mayor para `cold_start_max_interactions` o `cold_start_max_duration`.

HRNN-Coldstart presenta los siguientes límites de elementos fríos:

- `Maximum cold start items`: 80 000
- `Minimum cold start items`: 100

Si el número de elementos fríos está fuera de este rango, se producirá un error en los intentos de crear una solución.

La receta HRNN-Coldstart incluye las siguientes propiedades:

- Nombre – `aws-hrnn-coldstart`
- Nombre de recurso de Amazon (ARN) de la receta: `arn:aws:personalize:::recipe/aws-hrnn-coldstart`
- ARN del algoritmo: `arn:aws:personalize:::algorithm/aws-hrnn-coldstart`
- ARN de transformación de características: `arn:aws:personalize:::feature-transformation/featurize_coldstart`
- Tipo de receta: `USER_PERSONALIZATION`

Para obtener más información, consulte Elección de una receta.

En la siguiente tabla se describen los hiperparámetros de la receta HRNN-Coldstart. Un hiperparámetro es un parámetro de algoritmo que se puede ajustar para mejorar el rendimiento del modelo. Los hiperparámetros de algoritmos controlan el rendimiento del modelo. Los hiperparámetros de caracterización controlan cómo se filtran los datos que se van a utilizar en el entrenamiento. El proceso de elección del mejor valor para un hiperparámetro se denomina optimización de hiperparámetros (HPO). Para obtener más información, consulte Hiperparámetros y HPO.

La tabla también proporciona la siguiente información para cada hiperparámetro:

- Rango: [límite inferior, límite superior]
- Tipo de valor: número entero, continuo (flotante), categórico (booleano, lista, cadena)
- HPO ajustable: ¿puede el parámetro participar en HPO?

Nombre	Descripción
Hiperparámetros de algoritmos	
`hidden_dimension`	El número de variables ocultas utilizadas en el modelo. Las variables ocultas recrean el historial de compras de los usuarios y las estadísticas de los elementos para generar puntuaciones de clasificación. Especifiq ue un número mayor de dimensiones ocultas cuando el conjunto de datos de interacciones de elementos

Nombre	Descripción
	incluya patrones más complicados. El uso de más dimensiones ocultas requiere un conjunto de datos mayor y más tiempo de procesamiento. Para decidir el valor óptimo, utilice HPO. Para utilizar HPO, establezca `performHPO` en `true` cuando llame a las operaciones CreateSolution y CreateSolutionVersion. Valor predeterminado: 149 Rango: [32, 256] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`bptt`	Determina si se debe utilizar la propagación hacia atrás a través de la técnica de tiempo. La propagación hacia atrás a través del tiempo es una técnica que actualiza ponderaciones en algoritmos basados en redes neuronales recurrentes. Utilice `bptt` para créditos a largo plazo para conectar recompensas retrasadas a eventos tempranos. Por ejemplo, una recompensa retrasada puede ser una compra realizada después de varios clics. Un evento temprano puede ser un clic inicial. Incluso dentro de los mismos tipos de eventos, tales como un clic, es buena idea tener en cuenta los efectos a largo plazo y maximizar las recompensas totales. Para tener en cuenta los efectos a largo plazo, utilice valores mayores de `bptt`. El uso de un valor de `bptt` mayor requiere conjuntos de datos más grandes y más tiempo de procesamiento. Valor predeterminado: 32 Rango: [2, 32] Tipo de valor: entero HPO ajustable: sí

Nombre	Descripción
`recency_mask`	Determina si el modelo debe tener en cuenta las últimas tendencias de popularidad en el conjunto de datos de interacciones de elementos. Las últimas tendencias de popularidad pueden incluir cambios bruscos en los patrones subyacentes de los eventos de interacción. Para entrenar un modelo que tenga más peso sobre los eventos recientes, establezca `recency_mask` en `true`. Para entrenar un modelo que pesa igualmente todas las interacciones anteriores, establezca `recency_mask` en `false`. Para obtener buenas recomendaciones con el mismo peso, es posible que necesite un conjunto de datos de entrenamiento mayor. Valor predeterminado: #: `True` Rango: `True` o `False` Tipo de valor: booleano HPO ajustable: sí
Hiperparámetros de caracterización	
`cold_start_max_interactions`	El número máximo de interacciones de usuario-elemento que puede tener un elemento para que se considere un elemento frío. Valor predeterminado: 15 Rango: enteros positivos Tipo de valor: entero HPO ajustable: no

Nombre	Descripción
`cold_start_max_duration`	La duración máxima en días en relación con el punto de partida para que una interacción usuario-elemento se considere un elemento de arranque en frío. Para establecer el punto de partida de la interacción usuario-elemento, establezca el hiperparámetro `cold_start_relative_from` . Valor predeterminado: 5,0 Rango: flotantes positivos Tipo de valor: flotante HPO ajustable: no
`cold_start_relative_from`	Determina el punto de partida para que la receta HRNN-Coldstart calcule `cold_start_max_duration` . Para calcular a partir de la hora actual, elija `currentTime` . Para calcular `cold_start_max_duration` a partir de la marca de tiempo del último elemento en el conjunto de datos de interacciones de elementos, elija `latestItem` . Esto resulta útil si añade elementos nuevos con frecuencia. Valor predeterminado: #: `latestItem` Rango: `currentTime` , `latestItem` Tipo de valor: cadena HPO ajustable: no

Nombre	Descripción
`min_user_history_length_percentile`	El percentil mínimo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `min_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial cortas. Los usuarios con un historial corto suelen mostrar patrones basados en la popularidad de los elementos en lugar de sus necesidades o deseos personales. Eliminarlos puede entrenar modelos que se centren más en los patrones subyacentes de los datos. Elija un valor adecuado después de revisar las longitudes del historial de usuarios, utilizando un histograma o una herramienta similar. Recomendamos establecer un valor que retenga a la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,0 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Nombre	Descripción
`max_user_history_length_percentile`	El percentil máximo de longitudes de historial de usuario para incluir en el entrenamiento de modelos. La longitud del historial es la cantidad total de datos sobre un usuario. Utilice `max_user_history_length_percentile` para excluir un porcentaje de usuarios con longitudes de historial largas, ya que los datos de estos usuarios tienden a contener ruido. Por ejemplo, un robot podría tener una larga lista de interacciones automatizadas. La eliminación de estos usuarios limita el ruido en el entrenamiento. Elija un valor adecuado después de revisar las longitudes del historial de usuarios mediante un histograma o una herramienta similar. Recomendamos establecer un valor que retenga la mayoría de los usuarios, pero que elimine los casos de borde. Por ejemplo, al establecer `min__user_history_length_percentile to 0.05` y `max_user_history_length_percentile to 0.95` se incluyen todos los usuarios, excepto los que tienen longitudes de historial en el 5 % inferior o superior. Valor predeterminado: 0,99 Rango: [0,0, 1,0] Tipo de valor: flotante HPO ajustable: no

Preparación de los datos de entrenamiento para Amazon Personalize

Después de elegir un caso de uso de dominio o una receta y anotar los requisitos de datos, podrá empezar a preparar los datos. Amazon Personalize puede utilizar los siguientes tipos de datos:

- Interacciones de elemento: en Amazon Personalize, una interacción de elemento es un evento de interacción positiva entre un usuario y un elemento del catálogo. Por ejemplo, un usuario ve una película, ve un anuncio o compra un par de zapatos.
- Elementos: los metadatos de elemento pueden incluir información como el precio, el tipo de SKU, la descripción o la disponibilidad cada elemento del catálogo.
- Usuarios: los metadatos de usuario pueden incluir información como la edad, el sexo, el grado de fidelización y los intereses de cada uno de los usuarios.
- Acciones: una acción es una actividad de interacción que puede que quiera recomendar a los clientes. Entre las acciones puede incluirse instalar su aplicación móvil, completar un perfil de pertenencia a grupo, unirse a un programa de fidelización o registrarse para recibir correos electrónicos promocionales. Para la receta Next-Best-Action, se requiere el conjunto de datos de acciones. Ninguna otra receta personalizada ni caso de uso de dominio utiliza datos de acciones.
- Interacciones de acciones: una interacción de acciones es una evento de interacción entre un usuario y una acción. La receta Next-Best-Action utiliza estos datos y los de su conjunto de datos de acciones para recomendar acciones a los usuarios. Ninguna otra receta personalizada ni caso de uso de dominio utiliza datos de Action-interactions.

Amazon Personalize almacena los datos en conjuntos de datos, uno para cada tipo de datos. Cada conjunto de datos tiene requisitos diferentes. Al importar datos en un conjunto de datos de Amazon Personalize, puede optar por importar los registros de forma masiva, individualmente o de ambos modos. Las importaciones masivas implican la importación de una gran cantidad de registros históricos almacenados en uno o más archivos CSV de un bucket de Amazon S3.

- Si no tiene datos masivos, puede utilizar operaciones de importación individuales para recopilar datos y transmitir eventos hasta que cumpla los requisitos de entrenamiento de Amazon Personalize y los requisitos de datos del caso de uso o receta del dominio. Para obtener información sobre el registro de eventos, consulte Registro de los eventos en tiempo real para influir en las recomendaciones. Para obtener información sobre la importación de registros

individuales, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize.

- Si no está seguro de tener suficientes datos o si tiene dudas sobre la calidad, puede importar los datos a un conjunto de datos de Amazon Personalize y utilizar Amazon Personalize para analizarlos. Para obtener más información, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

En las siguientes secciones, se proporcionan los requisitos de datos para cada tipo de conjunto de datos de Amazon Personalize y las directrices para preparar datos masivos. Si no tiene datos masivos, consulte las secciones para comprender los datos obligatorios y opcionales que puede importar con operaciones de importación individuales. Si necesita ayuda adicional para dar formato a los datos, puede usar Amazon SageMaker AI Data Wrangler (Data Wrangler) para preparar los datos. Para obtener más información, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Temas

- Directrices de formato de datos masivos para todos los tipos de datos
- Preparación de los datos de interacción de elemento para el entrenamiento
- Preparación de los metadatos de elemento para el entrenamiento
- Preparación de los metadatos de usuario para el entrenamiento
- Preparación de los metadatos de acción para el entrenamiento
- Preparación de los datos de interacción de acción para el entrenamiento

Directrices de formato de datos masivos para todos los tipos de datos

Las siguientes directrices y requisitos pueden ayudarle a asegurarse de que los datos masivos tengan el formato correcto.

- Sus datos de entrada deben estar en un archivo CSV (valores separados por comas).

- La primera fila del archivo CSV debe contener sus encabezados de columnas. No incluya los encabezados entre comillas (").
- Las columnas deben tener nombres alfanuméricos únicos. Por ejemplo, no puede agregar un campo `GENRES_FIELD_1` y un campo `GENRESFIELD1` a la vez.
- Si va a importar varios archivos CSV, todos los encabezados de las columnas deben coincidir en todos los archivos.
- Asegúrese de tener los campos obligatorios para el tipo de conjunto de datos y asegúrese de que sus nombres cumplan con los requisitos de Amazon Personalize. Por ejemplo, sus datos de elementos podrían tener una columna llamada `ITEM_IDENTIFICATION_NUMBER` con los ID de cada uno de sus elementos. Para usar esta columna como un campo ITEM_ID, cambie el nombre de la columna a `ITEM_ID`. Si usa Data Wrangler para aplicar formato a sus datos, puede usar la transformación de Data Wrangler Asignar columnas para Amazon Personalize para asegurarse de que sus columnas tengan el nombre correcto.

  Para obtener información acerca del uso de Data Wrangler para preparar sus datos, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.
- Cada registro de su archivo CSV debe estar en una sola línea.
- Amazon Personalize no admite tipos de datos complejos, como matrices y mapas.
- Para que Amazon Personalize utilice datos booleanos al entrenar o filtrar, utilice valores de cadena `"True"` y `"False"`, o valores numéricos `1` para true y `0` para false.
- Si usa Data Wrangler para aplicar formato a sus datos, puede usar la transformación de Data Wrangler Analizar valor como tipo para convertir los tipos de datos.
- Los datos de `TIMESTAMP` y `CREATION_TIMESTAMP` deben estar en formato de tiempo UNIX. Para obtener más información, consulte Datos de marca temporal.
- Evite incluir caracteres `''` o caracteres especiales en los datos del ID de elemento, el ID de usuario y el ID de acción.
- Si sus datos incluyen caracteres codificados que no son ASCII, su archivo CSV debe estar codificado en formato UTF-8.
- Asegúrese de aplicar formato a los datos textuales tal y como se describe en Metadatos de texto no estructurado.

Preparación de los datos de interacción de elemento para el entrenamiento

Una interacción de elemento es una interacción positiva entre un usuario y un elemento del catálogo. Por ejemplo, un usuario ve una película, ve un anuncio o compra un par de zapatos. Debe importar los datos sobre las interacciones de los usuarios con sus elementos en un conjunto de datos de interacciones de elementos. Puede registrar varios tipos de eventos, como clic, ver o compra.

Por ejemplo, si un usuario hace clic en un elemento concreto y, a continuación, da a me gusta en el elemento, puede hacer que Amazon Personalize utilice estos eventos como datos de entrenamiento. Para cada evento, registraría el ID del usuario, el ID del elemento, la marca de tiempo (en formato de tiempo Unix) y el tipo de evento (hacer clic y me gusta). A continuación, se agregarían ambos eventos de interacción de elementos a un conjunto de datos de interacciones de elementos.

En todos los casos de uso de dominio y recetas personalizadas, los datos de interacciones de elemento masivos deben estar en un archivo CSV. Cada fila debe representar una interacción individual entre un usuario y un elemento. Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

En las siguientes secciones se proporciona más información sobre cómo preparar los datos de interacción de elemento para Amazon Personalize. Para ver las directrices de formato de datos masivos para todos los tipos de datos, consulte las directrices de formato de datos masivos

Temas

- Requisitos de datos de interacciones de elemento
- Datos de marca temporal
- Datos de valor de evento y tipo de evento
- Metadatos contextuales
- Datos de impresiones
- Ejemplo de datos de interacciones

Requisitos de datos de interacciones de elemento

En las siguientes secciones se enumeran los requisitos de datos de interacción de elemento para Amazon Personalize. Para conocer las cuotas adicionales, consulte Puntos de conexión y cuotas de Amazon Personalize.

Requisitos de entrenamiento mínimos

En todos los casos de uso de dominio y recetas personalizadas, los datos de interacciones de elemento masivos deben tener lo siguiente:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Como mínimo, 25 ID de usuarios únicos con al menos dos interacciones de elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

Para crear un recomendador o una solución personalizada, debe crear como mínimo un conjunto de datos de interacciones de elementos.

Requisitos de columna

El conjunto de datos de interacciones de elemento debe tener las siguientes columnas.

- USER_ID: identificador único del usuario que ha interactuado con el elemento. Cada evento debe tener un USER_ID. Debe ser una `string` con una longitud máxima de 256 caracteres.
- ITEM_ID: identificador único del elemento con el que ha interactuado el usuario. Cada evento debe tener un ID de elemento. Debe ser una `string` con una longitud máxima de 256 caracteres.
- TIMESTAMP: hora a la que se ha producido el evento (en formato de tiempo Unix en segundos). Cada interacción debe tener TIMESTAMP. Para obtener más información, consulte Datos de marca temporal.
- EVENT_TYPE: la naturaleza del evento de interacción con un elemento, como hacer clic, ver o comprar. Para los recomendadores de dominios, debe tener una columna de tipo de evento y cada interacción debe tener un tipo de evento. Para todas las recetas personalizadas, se recomienda

una columna EVENT_TYPE, pero es opcional. Si lo agrega, cada evento debe tener un tipo de evento. Para obtener más información consulte () Datos de valor de evento y tipo de evento.

Tiene la libertad de agregar campos personalizados adicionales según el caso de uso y los datos. El número máximo de columnas de metadatos opcionales es 5. Estas columnas pueden incluir valores vacíos o nulos. Se recomienda que estas columnas estén completadas en un 70 % como mínimo.

Datos de marca temporal

Los datos de marca temporal deben estar en formato de tiempo Unix expresado en segundos. Por ejemplo, la marca temporal de Epoch en segundos para la fecha del 31 de julio de 2020 es 1596238243. Para convertir fechas en marcas temporales de tiempo Unix, utilice un convertidor de tiempo, convertidor de marcas temporales Unix.

Amazon Personalize utiliza datos de marca temporal para calcular si son recientes e identificar cualquier patrón basado en el tiempo. Ayuda a Amazon Personalize a mantener las recomendaciones actualizadas con las preferencias en constante evolución de los usuarios.

Datos de valor de evento y tipo de evento

Un conjunto de datos de interacciones de elemento puede almacenar datos de tipo y valor de evento para cada interacción. Solo los recursos personalizados utilizan datos de valores de eventos.

Datos de tipo de evento

El tipo de evento de una interacción de elemento proporciona contexto sobre la naturaleza e importancia. Algunos ejemplos de tipos de eventos pueden ser hacer clic, ver o comprar. Amazon Personalize utiliza datos de tipo de evento, como datos de hacer clic o comprar, para identificar la intención y el interés del usuario. El número máximo de tipos de evento distintos combinados con el número total de columnas de metadatos opcionales en un conjunto de datos de interacciones de elemento es de 10.

Para los recomendadores de dominios, debe tener una columna de tipo de evento y cada interacción debe tener un tipo de evento. Para todas las recetas personalizadas, se recomienda una columna EVENT_TYPE, pero es opcional. Si lo agrega, cada evento debe tener un tipo de evento.

Si crea recursos personalizados, puede elegir los eventos que se utilizarán para el entrenamiento por tipo de evento. Si el conjunto de datos tiene varios tipos de evento en una columna EVENT_TYPE y no proporciona ningún tipo de evento al configurar una solución personalizada, Amazon Personalize utiliza todos los datos de interacciones con artículos para el entrenamiento con la misma

ponderación, con independencia del tipo. Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento.

Si tiene varios tipos de eventos y utiliza la receta User-Personalization-v2 o Personalized-Ranking-v2, puede especificar ponderaciones distintas para los diferentes tipos al configurar una solución personalizada. Por ejemplo, puede configurar una solución para dar más peso a los eventos de compra que a los de clic. Para obtener más información, consulte Optimización de una solución con la configuración de eventos.

Los siguientes casos de uso tienen requisitos de tipo de evento específicos:

Casos de uso de dominio de VIDEO_ON_DEMAND

- Como ha visto X, se requieren un mínimo de 1000 eventos `Watch`.
- Los más populares requieren un mínimo de 1000 eventos `Watch`.

Casos de uso de dominio de ECOMMERCE

- Los que más se han visto requieren un mínimo de 1000 eventos `View`.
- Los más vendidos requieren un mínimo de 1000 eventos `Purchase`.

Tipos de eventos positivos y negativos

Amazon Personalize supone que cualquier interacción es positiva. Las interacciones con un tipo de evento negativo, como no me gusta, no impedirán necesariamente que el elemento aparezca en las recomendaciones futuras del usuario.

A continuación, se indican formas de hacer que los eventos negativos y el desinterés de los usuarios influyan en las recomendaciones:

- Para todos los casos de uso de dominio y la receta User-Personalization, Amazon Personalize puede utilizar los datos de impresiones. Cuando un elemento aparece en los datos de impresiones y un usuario no los elige, es menos probable que aparezca en las recomendaciones. Para obtener más información, consulte Datos de impresiones.
- Si utiliza recursos personalizados e importa tipos de eventos positivos y negativos, puede centrarse únicamente en los tipos de eventos positivos y, después, filtrar los elementos con los que el usuario haya interactuado de forma negativa. Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento y Recomendaciones de filtrado y segmentos de usuarios.

Datos de valor de evento (recursos personalizados)

Los datos de valor de evento pueden ser el porcentaje de una película que ha visto un usuario o una valoración sobre 10. Si crea soluciones personalizadas, puede elegir los registros que se utilizan para el entrenamiento según los datos de las columnas EVENT_TYPE y EVENT_VALUE. Con los recomendadores de dominios, Amazon Personalize no usa datos de valor de eventos y no puede filtrar eventos antes del entrenamiento.

Para elegir registros según el tipo y el valor, registre los datos de tipo y valor de evento de los eventos. No todos los eventos deben tener un valor de evento. El valor que elija para cada evento depende de los datos que desee excluir y de los tipos de eventos que vaya a registrar. Por ejemplo, puede hacer coincidir la actividad del usuario, como el porcentaje de vídeo que ha visto el usuario, para los tipos de eventos ver.

Al configurar una solución, se establece un valor específico como umbral para excluir los registros del entrenamiento. Por ejemplo, si los datos EVENT_VALUE de los eventos con un EVENT_TYPE de ver son el porcentaje de un vídeo que ha visto un usuario, si establece el umbral del valor del evento en 0,5 y el tipo de evento en ver, Amazon Personalize entrena el modelo utilizando únicamente eventos de interacción ver con un EVENT_VALUE superior o igual a 0,5.

Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento

Metadatos contextuales

Con determinadas recetas y casos de uso de generadores de recomendaciones, Amazon Personalize puede utilizar metadatos contextuales para identificar los patrones subyacentes que revelan los elementos más relevantes para sus usuarios. Los metadatos contextuales son datos de interacciones que se recopilan en el entorno del usuario en el momento de un evento, como su ubicación o el tipo de dispositivo. También puede especificar el contexto de un usuario cuando reciba recomendaciones para el usuario.

Incluya metadatos contextuales para ofrecer una experiencia más personalizada a los usuarios y reducir la fase de arranque en frío para los nuevos usuarios. La fase de arranque en frío se produce cuando las recomendaciones son menos relevantes debido a la falta de datos de usuario históricos.

Por ejemplo, si el archivo CSV de interacciones de elemento incluye una columna DEVICE_TYPE con los valores `tablet` y `phone`, Amazon Personalize puede aprender cómo los clientes compran de forma diferente en dispositivos distintos. Cuando reciba recomendaciones para un usuario, puede

especificar el dispositivo y las recomendaciones serán más relevantes, aunque el usuario no tenga un historial de interacciones.

A continuación, se muestra cómo formatear un archivo CSV de interacciones de elemento con una columna DEVICE_TYPE como metadatos contextuales.

```
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
.....
```

Para los grupos de conjuntos de datos de dominio, los siguientes casos de uso de generador de recomendaciones pueden utilizar metadatos contextuales:

- Recomendaciones (dominio de ECOMMERCE)
- Mejores opciones (dominio de VIDEO_ON_DEMAND)

En el caso de los recursos personalizados, entre las recetas que utilizan metadatos contextuales se incluyen las siguientes:

- User-Personalization-v2 y User-Personalization
- Personalized-Ranking-v2 y Personalized-Ranking

Para obtener información sobre cómo incluir el contexto al recibir recomendaciones, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales. Para ver un ejemplo integral que muestra cómo usar metadatos contextuales, consulte la siguiente entrada del blog de AWS Machine Learning: Increasing the relevance of your Amazon Personalize recommendations by leveraging contextual information.

Datos de impresiones

Las impresiones son listas de elementos que estaban visibles para un usuario cuando interactuaba con un elemento concreto (por ejemplo, al hacer clic o verlo). Si utiliza un caso de uso de dominio que ofrezca personalización o la receta User-Personalization, Amazon Personalize puede utilizar los datos de impresiones para guiar la exploración.

Con la exploración, las recomendaciones incluyen algunos elementos o acciones que, por lo general, es menos probable que se recomienden al usuario, como nuevos elementos o acciones, elementos o acciones con pocas interacciones, o bien elementos o acciones menos relevantes para el usuario en función de su comportamiento anterior. Cuanto más frecuente sea la aparición de un elemento en los datos de impresiones, menos probabilidades hay de que Amazon Personalize incluya el elemento en la exploración.

Cuando crea un recomendador o una solución, Amazon Personalize siempre excluye los datos de impresiones del entrenamiento. Esto se debe a que Amazon Personalize no entrena los modelos con datos de impresiones. En vez de eso, los usa cuando recibe recomendaciones para guiar la exploración del usuario.

Los valores de impresión pueden tener como máximo 1000 caracteres (incluido el carácter de barra vertical). Para los grupos de conjuntos de datos de dominio, los siguientes casos de uso de recomendador pueden utilizar datos de impresiones:

- Recomendaciones (dominio de ECOMMERCE)
- Mejores opciones (dominio de VIDEO_ON_DEMAND)

Para obtener más información acerca de la exploración, consulte Exploration (Exploración). Amazon Personalize puede modelar dos tipos de impresiones: Impresiones implícitas y Impresiones explícitas.

Impresiones explícitas

Las impresiones explícitas son impresiones que registra y envía manualmente a Amazon Personalize. Use impresiones explícitas para manipular los resultados de Amazon Personalize. El orden de los elementos no tiene ningún impacto.

Por ejemplo, es posible que tenga una aplicación de compras que ofrezca recomendaciones de zapatos. Si solo recomienda zapatos que se encuentren en stock actualmente, puede especificar estos elementos utilizando impresiones explícitas. Su flujo de trabajo de recomendaciones usando impresiones explícitas podría ser el siguiente:

1. Solicite recomendaciones para uno de sus usuarios mediante la API de the section called “GetRecommendations” de API Amazon Personalize.
2. Amazon Personalize genera recomendaciones para el usuario que utiliza su modelo (versión de la solución) y las devuelve en la respuesta de la API.

3. Al usuario se muestran solo los zapatos recomendados que están en stock.
4. Para importar datos incrementales en tiempo real, cuando el usuario interactúa con un par de zapatos (por ejemplo, hace clic en ellos), registra la elección en una llamada a la API de PutEvents y enumera los elementos recomendados que se encuentran en stock en el parámetro `impression`. Para ver una muestra de código, consulte Registro de eventos de interacción de elemento con datos de impresiones.

   Para importar impresiones en los datos históricos de interacciones de elementos, puede enumerar las impresiones explícitas en su archivo csv y separar cada elemento con un carácter “|”. El carácter de la barra vertical cuenta para el límite de 1000 caracteres. Para ver un ejemplo, consulte Formato de impresiones explícitas.
5. Amazon Personalize utiliza los datos de impresiones para guiar la exploración, donde las recomendaciones futuras incluyen nuevos zapatos con menos datos de interacciones o relevancia.

Formato de impresiones explícitas

Para incluir impresiones explícitas en el archivo CSV, agregue una columna IMPRESSION. Para cada interacción de elemento, agregue una lista de ID de elementos separados por una barra vertical, “|”. El carácter de la barra vertical cuenta para el límite de 1000 caracteres de los datos de impresiones. Si incluye impresiones explícitas en la operación PutEvents, especifique los elementos en una matriz de cadenas.

A continuación, se ofrece un breve extracto de un archivo CSV que incluye impresiones explícitas en la columna `IMPRESSION`.

EVENT_TYPE	IMPRESSION	ITEM_ID	TIMESTAMP	USER_ID
click	73|70|17|95|96	73	1586731606	USUARIO_1
click	35|82|78|57|20|63|1|90|76|75|49|71|26|24|25|6	35	1586735164	USER_2
...	...	...	...	...

La aplicación mostró elementos de usuario de USER_1 73, 70, 17, 95 y 96, y el usuario finalmente eligió el elemento 73. Al crear una nueva versión de solución basada en estos datos, los elementos 70, 17, 95 y 96 serán recomendados con menor frecuencia al usuario USER_1.

Impresiones implícitas

Las impresiones implícitas son las recomendaciones, recuperadas de Amazon Personalize, que muestra al usuario. No es necesario que el archivo CSV incluya las columnas IMPRESSION o RECOMMENDATION_ID para utilizar impresiones implícitas. En vez de eso, incluya las `RecommendationId` (devueltas por las operaciones GetRecommendations y GetPersonalizedRanking) en las solicitudes de PutEvents. Amazon Personalize obtiene las impresiones implícitas en función de los datos de sus recomendaciones.

Por ejemplo, es posible que tenga una aplicación que proporcione recomendaciones para streaming en vídeo. El flujo de trabajo de recomendaciones que utiliza impresiones implícitas podría ser el siguiente:

1. Solicita recomendaciones de vídeo para uno de sus usuarios mediante la operación de la API de the section called “GetRecommendations” de Amazon Personalize.
2. Amazon Personalize genera recomendaciones para el usuario que utiliza su modelo (versión de la solución) y las devuelve con un `recommendationId` en la respuesta de la API.
3. Las recomendaciones de vídeo se muestran al usuario en su aplicación.
4. Cuando el usuario interactúa con un vídeo (por ejemplo, hace clic en él), registra la elección en una llamada a la API de PutEvents e incluye el `recommendationId` como parámetro. Para ver una muestra de código, consulte Registro de eventos de interacción de elemento con datos de impresiones.
5. Amazon Personalize utiliza el `recommendationId` para obtener los datos de impresión de las recomendaciones de vídeo anteriores y, a continuación, los utiliza para guiar la exploración, donde las recomendaciones futuras incluyen nuevos vídeos con menos datos de interacciones o relevancia.

   Para obtener más información sobre cómo registrar eventos con datos de impresiones implícitos, consulte Registro de eventos de interacción de elemento con datos de impresiones.

Ejemplo de datos de interacciones

Los siguientes datos de interacciones representan la actividad de usuario histórica de un sitio web que transmite vídeo. Puede usar los datos para entrenar un modelo que ofrezca recomendaciones de películas basadas en los datos de interacciones de los usuarios. Tenga en cuenta que algunos valores para EVENT_VALUE son nulos.

```
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 requiere las columnas `USER_ID`, `ITEM_ID` y `TIMESTAMP`. `USER_ID` es el identificador de un usuario de la aplicación. `ITEM_ID` es el identificador de una película. `EVENT_TYPE` y `EVENT_VALUE` son los identificadores de las interacciones de usuario. En los datos de muestra, los eventos son `watch` y `click` y los valores son el porcentaje de un vídeo que ha visto un usuario. `TIMESTAMP` representa la hora en formato Unix en el que se produjo la compra de la película.

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Este es el aspecto que tendría el archivo JSON de esquema para los datos de muestra.

```
{
  "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"
}
```

Preparación de los metadatos de elemento para el entrenamiento

Los metadatos de elemento incluyen datos numéricos y categóricos sobre los elementos con los que interactúan los usuarios. Entre los metadatos de elemento se incluyen la marca temporal de creación, el precio, el género, la descripción y la disponibilidad. Importa metadatos sobre sus elementos en un conjunto de datos de elementos de Amazon Personalize.

Según el caso de uso de dominio o la receta personalizada, los metadatos de elemento pueden ayudar a Amazon Personalize a recomendar elementos más relevantes a los usuarios, a predecir elementos similares de forma más precisa o a recomendar segmentos de usuarios más significativos. Y puede ayudar a Amazon Personalize a incluir nuevos elementos en las recomendaciones. Los metadatos de elemento son obligatorios para algunos casos de uso de dominio y son opcionales para todas las recetas personalizadas. Para obtener más información, consulte los requisitos de datos para el caso de uso de dominio o receta en Correlación del caso de uso con los recursos de Amazon Personalize.

Cuando se realiza el entrenamiento, Amazon Personalize no usa datos de elementos de cadena no categóricos, como títulos de elemento o datos de autor. Sin embargo, la importación de estos datos aún puede mejorar las recomendaciones. Para obtener más información, consulte Datos de cadena no categóricos.

El número máximo de elementos que Amazon Personalize tiene en cuenta durante el entrenamiento depende del caso de uso o la receta. Solo los elementos que se tienen en cuenta durante el entrenamiento pueden aparecer en las recomendaciones.

- Para User-Personalization-v2 o Personalized-Ranking-v2, el número máximo de elementos que se tienen en cuenta en un modelo durante el entrenamiento es de cinco millones. Estos elementos provienen del conjunto de datos de elementos y de interacciones de elemento.
- Para todos los casos de uso de dominio y recetas personalizadas distintos de User-Personalization-v2 y Personalized-Ranking-v2, el número máximo de elementos que un modelo tiene en cuenta durante el entrenamiento y la generación de recomendaciones es de 750 000.

En todos los casos de uso de dominio y recetas personalizadas, los datos de elemento masivos deben estar en un archivo CSV. Cada fila del archivo debe representar un elemento único. Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

En las siguientes secciones se proporciona más información sobre cómo preparar los metadatos de elemento para Amazon Personalize. Para ver las directrices de formato de datos masivos para todos los tipos de datos, consulte las directrices de formato de datos masivos

Temas

- Requisitos de datos de elemento
- Datos de la marca de tiempo de creación
- Metadatos categóricos
- Metadatos de texto no estructurado
- Datos numéricos
- Datos de cadena no categóricos
- Ejemplo de metadatos de elementos

Requisitos de datos de elemento

A continuación, se indican los requisitos de metadatos de elemento para Amazon Personalize.

Si no está seguro de tener suficientes datos o si tiene dudas sobre la calidad, puede importar los datos a un conjunto de datos de Amazon Personalize y utilizar Amazon Personalize para analizarlos.

Para obtener más información, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

- Para todos los casos de uso de dominio y recetas personalizadas, debe tener una columna ITEM_ID que almacene el identificador único de cada elemento. Cada elemento debe tener un ID de elemento. Debe ser una `string` con una longitud máxima de 256 caracteres.
- Para las recetas personalizadas, los datos deben tener al menos una columna de cadenas categóricas o de metadatos numéricos. Las columnas de metadatos de elemento pueden incluir valores vacíos o nulos. Se recomienda que estas columnas estén completadas en un 70 % como mínimo.
- Para los casos de uso de dominio, las columnas obligatorias dependen del dominio. Para obtener más información, consulte Requisitos de dominio de VIDEO_DEMAND o Requisitos de dominio de ECOMMERCE.
- El número máximo de columnas de metadatos es 100.

Requisitos de dominio de VIDEO_DEMAND

Para algunos casos de uso, se requieren metadatos de elemento (consulte Casos de uso de VIDEO_ON_DEMAND). Si es opcional, recomendamos importar metadatos de elemento para obtener las recomendaciones más relevantes. Si importa metadatos de elemento, los datos deben incluir las siguientes columnas:

- ITEM_ID
- GENRES (`string` categórica)
- CREATION_TIMESTAMP (en formato de tiempo Unix)

A continuación, se enumeran las columnas recomendadas adicionales y los tipos obligatorios. El tipo `null` indica que es posible que falten valores en la columna. Se recomienda que estas columnas estén completadas en un 70 % como mínimo. La inclusión de estas columnas puede mejorar las recomendaciones.

- PRICE (float)
- DURATION (float)
- GENERE_L2 (`string` categórica, `null`)
- GENRE_L3 (`string` categórica, `null`)

- AVERAGE_RATING (`float`, `null`)
- PRODUCT_DESCRIPTION (`string` textual, `null`)
- CONTENT_OWNER (`string` categórica, `null`): empresa propietaria del vídeo. Por ejemplo, los valores pueden ser HBO, Paramount y NBC.
- CONTENT_CLASSIFICATION (`string` categórica, `null`): valoración del contenido. Por ejemplo, los valores pueden ser G, PG, PG-13, R, NC-17 y no sin clasificación.

Requisitos de dominio de ECOMMERCE

Los metadatos de elemento son opcionales para todos los casos de uso de ECOMMERCE. Si tiene datos de elemento, recomendamos importarlos para obtener las recomendaciones más relevantes. Si importa metadatos de elemento, los datos deben tener las siguientes columnas:

- ITEM_ID
- PRICE (`float`)
- CATEGORY_L1 (`string` categórica): para obtener información sobre el formato de datos categóricos, consulte Metadatos categóricos.

A continuación, se enumeran las columnas recomendadas adicionales y los tipos obligatorios. El tipo `null` indica que es posible que falten valores en la columna. Se recomienda que estas columnas estén completadas en un 70 % como mínimo. La inclusión de estas columnas puede mejorar las recomendaciones.

- CATEGORY_L2 (`string`, categórica, `null`)
- CATEGORY_L3 (`string`, categórica, `null`)
- PRODUCT_DESCRIPTION (`string` textual, `null`)
- CREATION_TIMESTAMP (`float`)
- AGE_GROUP (`string` categórica, `null`): grupo de edad al que se destina el elemento. Los valores pueden ser recién nacidos, bebés, niños y adultos.
- ADULT (`string` categórica, `null`): indica si el elemento está restringido solo a adultos, por ejemplo, las bebidas alcohólicas. Los valores pueden ser sí o no.
- GENDER (`string` categórica, `null`): género al que va destinado el elemento. Los valores pueden ser hombre, mujer y unisex.

Datos de la marca de tiempo de creación

Los datos de marca temporal de creación deben estar en formato de tiempo Unix expresado en segundos. Por ejemplo, la marca temporal de Epoch en segundos para la fecha del 31 de julio de 2020 es 1596238243. Para convertir fechas en marcas temporales de tiempo Unix, utilice un convertidor de tiempo, convertidor de marcas temporales Unix.

Amazon Personalize utiliza los datos de marca temporal de creación (en formato de tiempo Unix, en segundos) para calcular la antigüedad de un elemento y ajustar las recomendaciones en consecuencia.

Si faltan los datos de marca de tiempo de creación para uno o más elementos, Amazon Personalize deduce esta información a partir de los datos de interacción, si los hay, y utiliza la marca de tiempo de los datos de interacción más antiguos del elemento como la marca de tiempo de creación del elemento. Si un elemento no tiene datos de interacción, su marca de tiempo de creación se establece como la marca de tiempo de la última interacción del conjunto de entrenamiento y Amazon Personalize lo considera un elemento nuevo.

Metadatos categóricos

Con determinadas recetas y casos de uso de dominio, Amazon Personalize utiliza metadatos categóricos, como el género o el color de un elemento, a la hora de identificar los patrones subyacentes que revelan los elementos más relevantes para los usuarios. Puede definir su propio rango de valores en función de su caso de uso. Los metadatos categóricos pueden estar en cualquier idioma.

Para los elementos con varias categorías, separe cada valor con la barra vertical, “|”. Por ejemplo, para un campo GENRES, los datos para un elemento podrían ser `Action|Crime|Biopic`. Si tiene varios niveles de datos categóricos y algunos elementos tienen varias categorías para cada nivel de la jerarquía, use un campo independiente para cada nivel y adjunte un indicador de nivel después del nombre de cada campo: GENRES, GENRE_L2, GENRE_L3. Esto le permite filtrar las recomendaciones en función de las subcategorías, incluso si un elemento pertenece a varias categorías de múltiples niveles (para obtener información sobre la creación y el uso de filtros, consulte Recomendaciones de filtrado y segmentos de usuarios). Por ejemplo, un vídeo puede tener los siguientes datos para cada nivel de categoría:

- GENRES: Acción|Aventura
- GENRE_L2: Crimen|Western
- GENRE_L3: Biopic

En este ejemplo, el vídeo está en la jerarquía acción > crimen > biopic y la jerarquía aventura > western > biopic. Recomendamos usar solo hasta L3, pero puede usar más niveles si es necesario.

Los valores categóricos pueden tener una longitud máxima de 1000 caracteres. Si tiene un elemento con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos. Recomendamos que las columnas categóricas tengan un máximo de 1000 valores posibles. La importación de datos categóricos con más valores puede afectar negativamente a las recomendaciones. Lo siguiente puede ayudarlo a reducir el número de valores posibles de una columna categórica:

- Asegúrese de que los valores sigan una convención de nomenclatura coherente y compruebe si hay errores tipográficos. Por ejemplo, use “Zapatos de hombre” en lugar de tener una combinación de “Zapatos de hombre”, “Zapatos para hombre” y “Calzado de hombre”.
- Consolide categorías similares que usen términos ligeramente diferentes que se refieran a la misma categoría subyacente, como “Zapatos” y “Zapatillas”.
- Si los datos tienen una estructura jerárquica, en la que las categorías más amplias (como “Calzado”) contienen subcategorías más específicas (como “Calzado de hombre”, “Calzado de mujer” o “Calzado de niño”), utilice una columna independiente para cada nivel y agregue un indicador de nivel después del nombre de cada campo. Por ejemplo, CATEGORY_1, CATEGORY_2 y CATEGORY_3. Esto puede reducir las categorías ambiguas o superpuestas.

Con todas las recetas y dominios, puede importar datos categóricos y usarlos para filtrar las recomendaciones en función de los atributos de un elemento. Para obtener más información acerca del filtrado de recomendaciones, consulte Recomendaciones de filtrado y segmentos de usuarios.

Metadatos de texto no estructurado

Con determinadas recetas y dominios, Amazon Personalize puede extraer información significativa de metadatos de texto no estructurado, como descripciones de productos, reseñas de productos o sinopsis de películas. Amazon Personalize utiliza texto no estructurado para identificar los elementos relevantes para los usuarios, en particular cuando los elementos son nuevos o tienen menos datos de interacciones. Puede añadir 1 campo textual como máximo. Incluya datos de texto no estructurado en su conjunto de datos de elementos para aumentar las tasas de clics y las tasas de conversación para los nuevos elementos de su catálogo.

Cuando prepare los metadatos de texto no estructurado, escriba el texto entre comillas y elimine los caracteres de nueva línea. Utilice el carácter \ para aplicar escape en cualquier comilla doble

o carácter \ en los datos. Amazon Personalize trunca los campos de texto al límite de caracteres. Asegúrese de que la información más relevante del texto se encuentre al principio del campo.

Los valores de texto no estructurado pueden tener como máximo 20 000 caracteres en todos los idiomas, excepto en chino y japonés. Para el chino y el japonés, puede tener como máximo 7000 caracteres. Amazon Personalize trunca los valores que superan el límite de caracteres al límite de caracteres.

Puede enviar elementos de texto no estructurado en varios idiomas, pero el texto de cada elemento debe estar en un solo idioma. El texto puede estar en los siguientes idiomas:

- Chino simplificado
- Chino tradicional
- Inglés
- Francés
- Alemán
- Japonés
- Portugués
- Español

Datos numéricos

Amazon Personalize puede utilizar metadatos de elemento numéricos, como el precio o la duración del vídeo, a fin de generar recomendaciones más relevantes para los usuarios. Estos datos numéricos se pueden representar como números enteros o valores decimales.

Si usa las recetas personalizadas User-Personalization o Personalized-Ranking, puede optimizar una solución de Amazon Personalize para un objetivo relacionado con los metadatos de elemento junto con la máxima relevancia, como maximizar los ingresos. Al configurar la solución, elija la columna de metadatos numéricos del conjunto de datos de elementos que esté relacionada con el objetivo. Por ejemplo, puede elegir una columna VIDEO_LENGTH para maximizar los minutos de streaming o una columna PRICE para maximizar los ingresos.

Para obtener más información, consulte Optimización de una solución para un objetivo adicional.

Datos de cadena no categóricos

A excepción de los ID de elemento, Amazon Personalize no usa datos de cadenas no categóricas al realizar el entrenamiento, como títulos de elemento o datos de autor. Sin embargo, Amazon Personalize puede usarlos con las siguientes características. Los valores no categóricos pueden tener 1000 caracteres como máximo.

- Amazon Personalize puede incluir metadatos de los elementos en las recomendaciones, incluidos valores de cadena no categóricos. Puede usar los metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar el nombre del director al carrusel de recomendaciones de una película. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.
- Si usa Similar-Items, puede generar recomendaciones por lotes con temas. Al generar recomendaciones por lotes con temas, debe especificar una columna de nombre del elemento en el trabajo de inferencia por lotes. Para obtener más información, consulte Recomendaciones por lotes con temas del Generador de contenidos.
- Puede crear filtros para incluir o quitar elementos de las recomendaciones en función de los datos de cadena no categóricos. Para obtener más información acerca de los filtros, consulte Recomendaciones de filtrado y segmentos de usuarios.

Ejemplo de metadatos de elementos

Las primeras líneas de metadatos de películas de un archivo CSV pueden tener el siguiente aspecto.

```
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 columna `ITEM_ID` es obligatoria y almacena identificadores únicos para cada elemento individual. La columna `DESCRIPTION` almacena metadatos categóricos para cada película y la columna `GENRE` son metadatos textuales no estructurados. La columna `CREATION_TIMESTAMP` almacena la hora de creación de cada elemento en formato de tiempo Unix en segundos.

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Este es el aspecto que tendría el archivo JSON de esquema para los datos de ejemplo anteriores.

```
{
  "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"
}
```

Preparación de los metadatos de usuario para el entrenamiento

Los datos de usuario que puede importar en Amazon Personalize incluyen metadatos numéricos, como la edad de los usuarios, y categóricos, como el sexo o el grado de fidelización. Los metadatos sobre los usuarios se importan en un conjunto de datos de Usuarios de Amazon Personalize.

Según el caso de uso de dominio o la receta personalizada, los metadatos de usuario pueden ayudar a Amazon Personalize a recomendar elementos más relevantes a los usuarios o a recomendar segmentos de usuarios más significativos. Además, después del entrenamiento, pueden ayudar al modelo a recomendar elementos a los usuarios sin datos de interacciones. Para obtener más información sobre qué casos de uso o recetas utilizan metadatos de usuario, consulte los requisitos de datos para el caso de uso de dominio o la receta en Correlación del caso de uso con los recursos de Amazon Personalize.

En el entrenamiento, Amazon Personalize no usa datos de usuario de cadena no categóricos, como nombres de usuario, palabras clave sobre el usuario o etiquetas. Sin embargo, la importación de estos datos aún puede mejorar las recomendaciones. Para obtener más información, consulte Datos de cadena no categóricos.

En todos los casos de uso de dominio y recetas personalizadas, los datos de usuario masivos deben estar en un archivo CSV. Cada fila del archivo debe representar a un usuario único. Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

En las siguientes secciones se proporciona más información sobre cómo preparar los datos de usuario para Amazon Personalize. Para ver las directrices de formato de datos masivos para todos los tipos de datos, consulte las directrices de formato de datos masivos

Temas

- Requisitos de datos de usuario
- Metadatos categóricos
- Datos de cadena no categóricos
- Ejemplo de metadatos de usuarios

Requisitos de datos de usuario

A continuación, se indican los requisitos de datos de usuario para Amazon Personalize. Tiene la libertad de agregar campos personalizados adicionales según el caso de uso y los datos.

- Los datos deben tener una columna USER_ID que almacene el identificador único de cada usuario. Cada usuario debe tener un ID de usuario. Debe ser una `string` con una longitud máxima de 256 caracteres.
- Los datos deben tener al menos una columna de cadenas categóricas o de metadatos numéricos. Las columnas de metadatos de usuario pueden incluir valores vacíos o nulos para algunos usuarios. Se recomienda que estas columnas estén completadas en un 70 % como mínimo.
- El número máximo de columnas de metadatos es 25.

Si no está seguro de tener suficientes datos o si tiene dudas sobre la calidad, puede importar los datos a un conjunto de datos de Amazon Personalize y utilizar Amazon Personalize para analizarlos. Para obtener más información, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

Metadatos categóricos

Con algunas recetas y todos los casos de uso de dominio, Amazon Personalize utiliza metadatos categóricos, como el sexo del usuario, los intereses o el estado de pertenencia, para identificar los patrones subyacentes que revelan los elementos más relevantes para los usuarios. Puede definir su propio rango de valores en función de su caso de uso. Los metadatos categóricos pueden estar en cualquier idioma.

Para los usuarios con varias categorías, separe cada valor con la barra vertical, “|”. Por ejemplo, en el caso de un campo INTERESTS, los datos de un usuario pueden ser `Movies|TV Shows|Music`.

Con todas las recetas y dominios, puede importar metadatos categóricos y utilizarlos para filtrar las recomendaciones en función de los atributos de un usuario. Para obtener información acerca del filtrado de recomendaciones, consulte Recomendaciones de filtrado y segmentos de usuarios.

Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un usuario con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Datos de cadena no categóricos

A excepción de los ID de usuario, Amazon Personalize no usa datos de cadena no categóricos al realizar el entrenamiento, como nombres de usuario, palabras clave sobre el usuario o etiquetas. Sin embargo, Amazon Personalize puede usarlos al filtrar recomendaciones. Puede crear filtros para incluir o eliminar elementos de las recomendaciones en función de los datos de cadena no categóricos sobre el usuario para el que obtiene recomendaciones (CurrentUser). Para obtener más información acerca de los filtros, consulte Recomendaciones de filtrado y segmentos de usuarios. Los valores no categóricos pueden tener 1000 caracteres como máximo.

Ejemplo de metadatos de usuarios

Las primeras líneas de metadatos de usuario de un archivo CSV pueden tener el siguiente aspecto.

```
USER_ID,AGE,GENDER,INTEREST
5,34,Male,hiking
6,56,Female,music
8,65,Male,movies|TV shows|music
...
...
```

La columna `USER_ID` es obligatoria y almacena identificadores únicos para cada usuario individual. La columna `AGE` son metadatos numéricos. Las columnas `GENDER` y `INTEREST` almacenan metadatos categóricos para cada usuario.

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Este es el aspecto que tendría el archivo JSON de esquema para los datos de ejemplo anteriores.

```
{
  "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"
}
```

Preparación de los metadatos de acción para el entrenamiento

Una acción es una actividad que genera ingresos o interacción y que puede que quiera recomendar a sus usuarios. Entre las acciones puede incluirse instalar su aplicación móvil, completar un perfil de pertenencia a grupo, unirse a un programa de fidelización o registrarse para recibir correos electrónicos promocionales. Los datos sobre sus acciones se importan en un conjunto de datos de acciones de Amazon Personalize. Entre los ejemplos de datos de una acción se incluyen el ID único de la acción, el valor estimado de la acción o la marca temporal de caducidad de esta.

Si usa Next-Best-Action, debe importar los metadatos de acción. Con esta receta, Amazon Personalize predice la siguiente mejor acción a partir de las acciones que importe en el conjunto de datos de acciones. Ninguna otra receta o caso de uso utiliza metadatos de acción. No puede crear un conjunto de datos de acciones en un grupo de conjuntos de datos de dominio.

Cuando se realiza el entrenamiento, Amazon Personalize no usa datos de acciones de cadena no categóricos, como títulos de acción o etiquetas. Sin embargo, la importación de estos datos aún puede mejorar las recomendaciones. Para obtener más información, consulte Datos de cadena no categóricos.

Los datos de acción por lotes deben estar en un archivo CSV. Cada fila del archivo debe representar una acción única. Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

En las siguientes secciones se proporciona más información sobre cómo preparar los metadatos de acción para Amazon Personalize. Para ver las directrices de formato de datos masivos para todos los tipos de datos, consulte las directrices de formato de datos masivos

Temas

- Requisitos de datos de acciones
- Datos de la marca de tiempo de caducidad de la acción
- Datos de frecuencia de repetición
- Value data
- Datos de la marca de tiempo de creación
- Metadatos categóricos
- Datos de cadena no categóricos
- Ejemplo de metadatos de acción

Requisitos de datos de acciones

A continuación, se indican los requisitos de datos de acciones para Amazon Personalize.

- Debe tener una columna ACTION_ID que almacene el identificador único de cada acción. Cada acción debe tener un ID de elemento. Debe ser una `string` con una longitud máxima de 256 caracteres.
- Los datos deben tener al menos una columna de cadenas categóricas o de metadatos numéricos. Las columnas de metadatos de acción pueden incluir valores vacíos o nulos. Se recomienda que estas columnas estén completadas en un 70 % como mínimo.
- Durante el entrenamiento de modelos, Amazon Personalize tiene en cuenta un máximo de 1000 acciones. Si importa más de 1000 acciones, Amazon Personalize decide cuáles incluir en el entrenamiento, para lo que se da prioridad a las acciones nuevas (acciones que ha agregado recientemente sin interacciones) y las acciones existentes con datos de interacciones recientes.
- El número máximo de columnas es 10.

Datos de la marca de tiempo de caducidad de la acción

La marca de tiempo de caducidad de una acción especifica la fecha en la que una acción deja de ser válida. Los datos de la marca de tiempo de caducidad de la acción se proporcionan en formato

de tiempo Unix, en segundos. Si una acción ha caducado, Amazon Personalize no la incluirá en las recomendaciones.

Especifique una marca de tiempo de caducidad para sus acciones si quiere limitar su aparición en las recomendaciones a un período de tiempo determinado. Por ejemplo, puede que tenga una aplicación que ejecute una campaña de suscripción durante un mes determinado. Puede establecer una marca de tiempo de caducidad para la acción de inscribirse al final de ese mes. Amazon Personalize deja de recomendar esta acción automáticamente al llegar a esa fecha.

Si establece la marca de tiempo de caducidad en una fecha pasada para una nueva acción, o si actualiza la marca de tiempo de una acción a una fecha pasada, pueden tardarse hasta 2 horas en quitar la acción de las recomendaciones.

Datos de frecuencia de repetición

Los datos de frecuencia de repetición especifican cuántos días debe esperar Amazon Personalize para recomendar una acción determinada después de la interacción de un usuario, según el historial del usuario en su conjunto de datos de interacciones de acción. La frecuencia de repetición de una acción se especifica en días y el máximo es de 30.

Por ejemplo, puede que tenga una aplicación de comercio electrónico en la que cada usuario crea una cuenta y un perfil. Si tiene una acción `complete profile` y quiere esperar una semana después de que un usuario interactúe con ella antes de volver a recomendarla, debe especificar 7 días como valor `REPEAT_FREQENCY` de la acción. Transcurridos 7 días, Amazon Personalize empieza a tener en cuenta la acción para las recomendaciones.

Si no se especifica la frecuencia de repetición de una acción, Amazon Personalize no establecerá ningún límite respecto al número de veces que esta aparece en las recomendaciones.

Value data

Los datos de valor son el valor empresarial o la importancia de cada acción. El valor `value` de una acción puede oscilar entre 1 y 10, donde 10 es la acción más valiosa del conjunto de datos.

Por ejemplo, supongamos que tiene dos acciones: una para inscribirse en la suscripción básica y otra para inscribirse en el servicio premium. Para el servicio básico, puede especificar un valor de `5` y, para el premium, un valor de `10`.

Amazon Personalize utiliza los datos de valor como una entrada al determinar la mejor acción para recomendar a los usuarios. Por ejemplo, si un usuario tiene la misma probabilidad de realizar una

acción u otra, Amazon Personalize clasifica la acción con el valor más alto en una posición superior en las recomendaciones.

Datos de la marca de tiempo de creación

Amazon Personalize utiliza los datos de la marca de tiempo de creación (en formato de tiempo Unix, en segundos) para calcular la antigüedad de una acción y ajustar las recomendaciones en consecuencia.

Si no tiene datos de marca de tiempo de creación, Amazon Personalize deduce esta información de los datos de interacción de la acción. Utiliza la marca de tiempo de los datos de interacción más antiguos de la acción como marca de tiempo de creación de la acción. Si una acción no tiene datos de interacción, su marca de tiempo de creación se establece como la marca de tiempo de la última interacción del conjunto de entrenamiento y Amazon Personalize la considera una acción nueva.

Metadatos categóricos

Amazon Personalize utiliza metadatos categóricos sobre las acciones, como la estacionalidad o la exclusividad de las acciones, al identificar los patrones subyacentes que revelan las mejores acciones para los usuarios. Puede definir su propio rango de valores en función de su caso de uso. Los metadatos categóricos pueden estar en cualquier idioma.

Puede importar datos categóricos y usarlos para filtrar las recomendaciones en función de los atributos de una acción. Para obtener más información acerca del filtrado de recomendaciones, consulte Recomendaciones de filtrado y segmentos de usuarios.

Los valores categóricos pueden tener una longitud máxima de 1000 caracteres. Si tiene una acción con un valor categórico con más de 1000 caracteres, se generará un error en el trabajo de importación del conjunto de datos.

Datos de cadena no categóricos

A excepción de los ID de acción, Amazon Personalize no usa datos de cadena no categóricos al realizar el entrenamiento, como el nombre de una acción, palabras clave sobre la acción o etiquetas. Sin embargo, Amazon Personalize puede usarlos al filtrar recomendaciones. Puede crear filtros para incluir o quitar acciones de las recomendaciones en función de los datos de cadena no categóricos. Para obtener más información acerca de los filtros, consulte Recomendaciones de filtrado y segmentos de usuarios. Los valores no categóricos pueden tener 1000 caracteres como máximo.

Ejemplo de metadatos de acción

Las primeras líneas de metadatos de acción de un archivo CSV pueden tener el siguiente aspecto.

```
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 columna `ACTION_ID` es obligatoria. La columna `MEMBERSHIP_LEVEL` es un campo de cadena categórico. Los campos `VALUE`, `CREATION_TIMESTAMP` y `REPEAT_FREQUENCY` son palabras clave reservadas con los tipos necesarios.

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Este es el aspecto que tendría el archivo JSON de esquema para los datos de ejemplo anteriores.

```
{
  "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"
}
```

Preparación de los datos de interacción de acción para el entrenamiento

Si usa la receta personalizada Next-Best-Action, Amazon Personalize utiliza los datos de interacciones de acción para identificar el interés de los usuarios y predecir las acciones que es más probable que realicen. Una interacción de acción es una interacción en la que participan un usuario y una acción del conjunto de datos de acciones. Por ejemplo, si cuenta con una acción para inscribirse en el conjunto de datos de acciones y un usuario la realiza, debe registrar el ID del usuario, el ID de la acción y la marca de tiempo, además de registrar `TAKEN` para el tipo de evento.

Importe las interacciones de acción en un conjunto de datos de interacciones de acción de Amazon Personalize. Puede importar eventos de interacción de acciones en bloque con un trabajo de importación de conjuntos de datos, o bien transmitirlos en tiempo real con la operación de la API PutActionInteractions. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio.

Los datos de las interacciones de acción por lotes deben estar en un archivo CSV. Cada fila del archivo debe representar una interacción única entre un usuario y una acción. Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon

Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

En las siguientes secciones se proporciona más información sobre cómo preparar los datos de interacción de acción para Amazon Personalize. Para ver las directrices de formato de datos masivos para todos los tipos de datos, consulte las directrices de formato de datos masivos.

Temas

- Requisitos de datos de interacciones de acción
- Datos de tipo de evento
- Ejemplo de datos de interacciones de acción

Requisitos de datos de interacciones de acción

No hay ningún requisito mínimo respecto a los datos de interacciones de acción. Le recomendamos que los importe para obtener recomendaciones de acciones de calidad. Si no tiene datos de interacción de acciones, puede crear un conjunto de datos de interacciones de este tipo vacío y registrar las interacciones de los clientes con las acciones mediante la operación de la API PutActionInteractions.

Los datos de las interacciones de acción deben tener como mínimo las siguientes columnas. Tiene la libertad de agregar campos personalizados adicionales según el caso de uso y los datos.

- USER_ID: identificador único del usuario que ha interactuado con el elemento. Cada evento debe tener un USER_ID. Debe ser una `string` con una longitud máxima de 256 caracteres.
- ACTION_ID: identificador único del elemento con el que ha interactuado el usuario. Cada evento debe tener un ID de elemento. Debe ser una `string` con una longitud máxima de 256 caracteres.
- TIMESTAMP: hora a la que se ha producido el evento (en formato de tiempo Unix en segundos). Cada interacción de acción debe tener un campo TIMESTAMP. Para obtener más información, consulte Datos de marca temporal.
- EVENT_TYPE: indica si la acción se ha realizado, no se ha realizado o se ha visualizado. Cada interacción de acción debe tener un tipo de evento. Para obtener más información, consulte Datos de tipo de evento.

Hasta que se importen los datos de interacción de acción, Amazon Personalize recomienda acciones sin personalización y las puntuaciones de propensión son de 0,0. Una acción tendrá una puntuación cuando tenga lo siguiente:

- Al menos 50 interacciones de acción con el tipo de evento TAKEN.
- Al menos 50 interacciones de acción con el tipo de evento NOT_TAKEN o VIEWED.

Estas interacciones de acción deben estar presentes en el entrenamiento de la versión de solución más reciente y deben producirse en un intervalo de seis semanas a partir de la última marca temporal de interacción del conjunto de datos de interacciones de acción.

Datos de tipo de evento

Amazon Personalize puede usar patrones en los datos de los tipos de eventos para identificar las acciones que es más probable que realicen los usuarios. Por ejemplo, si un cliente ignora con frecuencia una acción de suscripción por correo electrónico (que se indica con el tipo de evento NOT_TAKEN), Amazon Personalize puede ajustar las recomendaciones para incluir menos acciones de este tipo.

Solo puede usar los siguientes tipos de eventos de interacción de acciones. Amazon Personalize utiliza estos eventos para obtener información sobre el usuario y calcular qué acciones recomendar a continuación.

- Realizado: registre los eventos como Realizado cuando un usuario realice una acción recomendada.
- No realizado: registre los eventos como No realizado cuando el usuario decida deliberadamente no realizar la acción después de verla. Por ejemplo, si elige No al mostrarle la acción. Los eventos de tipo No realizado pueden indicar que el cliente no está interesado en la acción.
- Visto: registre los eventos como Visto cuando muestre una acción al usuario antes de que tome la decisión de realizarla o no. Amazon Personalize utiliza los eventos Visto para obtener información sobre los intereses de los usuarios. Por ejemplo, si un usuario ve una acción pero no la realiza, puede que al usuario no le interese esta acción en el futuro.

Ejemplo de datos de interacciones de acción

Las primeras líneas de un archivo CSV con datos de interacciones de acción y todas las columnas obligatorias pueden tener el siguiente aspecto.

```
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
...
...
```

Después de finalizar la preparación de los datos, podrá crear un archivo JSON de esquema. Este archivo informa a Amazon Personalize de la estructura de los datos. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Este es el aspecto que tendría el archivo JSON de esquema para los datos de ejemplo anteriores.

```
{

  "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"
}
```

Creación de archivos JSON de esquema para los esquemas de Amazon Personalize

Después de preparar los datos, podrá crear archivos JSON de esquema para cada tipo de datos que vaya a importar. Estos archivos describen la estructura y el contenido de los datos, incluidos los nombres de las columnas y los tipos de datos.

Los archivos JSON de esquema se utilizan cuando se crea un esquema de Amazon Personalize en Creación de un esquema y un conjunto de datos. En Amazon Personalize, un esquema es un recurso que permite a Amazon Personalize analizar los datos cuando los importa al conjunto de datos. Cree un esquema para cada conjunto de datos que utilice.

En el caso de los recursos personalizados, cada conjunto de datos tiene requisitos de esquema específicos. En el caso de los grupos de conjuntos de datos de dominio, el dominio que elija determina los requisitos de conjunto de datos y esquema. Cada dominio tiene un esquema predeterminado para cada tipo de conjunto de datos. Al crear un conjunto de datos, puede usar el esquema de dominio existente o crear uno nuevo mediante la modificación del esquema predeterminado existente. Use el esquema predeterminado como guía para saber qué datos importar para su dominio.

En las siguientes secciones, se proporcionan los requisitos personalizados y de dominio para crear un archivo JSON de esquema para cada tipo de conjunto de datos.

Temas

- Requisitos de formato de esquema
- Conjuntos de datos y esquemas VIDEO_ON_DEMAND
- Conjuntos de datos y esquemas de ECOMMERCE
- Esquemas y conjuntos de datos personalizados

Requisitos de formato de esquema

Cuando crea un esquema para un conjunto de datos de un grupo de conjuntos de datos de dominio o un grupo de conjuntos de datos personalizados, debe seguir estas directrices:

- Debe definir el esquema en formato Avro. Para obtener información sobre los tipos de datos de Avro que admitimos, consulte Tipos de datos de esquema.

- Un esquema tiene una clave de nombre cuyo valor debe coincidir con el tipo de conjunto de datos.
- Los campos de esquema pueden aparecer en cualquier orden, pero deben coincidir con el orden de los encabezados de columna correspondientes en su archivo CSV.
- Los esquemas deben ser archivos JSON sin estructuras anidadas. Por ejemplo, un campo no puede ser el principal de varios subcampos.
- Los esquemas de Amazon Personalize no admiten tipos complejos, como matrices y mapas.
- Los campos de esquema deben tener nombres alfanuméricos únicos. Por ejemplo, no puede agregar un campo `GENRES_FIELD_1` y un campo `GENRESFIELD1` a la vez.
- Debe definir campos obligatorios con sus tipos de datos obligatorios. Los campos de cadenas categóricas reservadas deben tener el atributo `categorical` establecido en `true`, mientras que los campos de cadenas reservadas no pueden ser categóricos. Las palabras clave no pueden estar en sus datos.
- Si agrega su propio campo de metadatos de tipo `string` y quiere que Amazon Personalize lo utilice durante el entrenamiento, debe incluir el atributo `categorical` o el atributo `textual` (solo los esquemas de elementos admiten campos con el atributo textual).
- Amazon Personalize no utiliza datos de tipo `boolean` al entrenar o filtrar recomendaciones. Para que Amazon Personalize utilice datos booleanos al entrenar o filtrar, utilice un campo de tipo Cadena y utilice los valores `"True"` y `"False"` de sus datos. También puede usar el tipo int o long, y los valores `0` y `1`.
- Los campos textuales deben ser del tipo `string` y deben tener el atributo `textual` establecido en `true`. Para obtener más información acerca de datos de texto no estructurados, consulte Metadatos de texto no estructurado.

Los conjuntos de datos de grupos de conjuntos de datos de dominio tienen requisitos adicionales según el dominio y el tipo de conjunto de datos. Los conjuntos de datos de grupos de conjuntos de datos personalizados tienen requisitos adicionales según el tipo.

Tipos de datos de esquema

Los esquemas de Amazon Personalize admiten los siguientes tipos de Avro para los campos:

- float
- double
- int
- long

- cadena
- booleano
- null

Algunos campos obligatorios y reservados admiten datos nulos. La adición de un tipo de `null` a un campo le permite usar datos imperfectos (por ejemplo, metadatos con valores en blanco) para generar recomendaciones. Para obtener información sobre qué campos admiten datos nulos, consulte el tema de requisitos de esquema para el dominio: Conjuntos de datos y esquemas VIDEO_ON_DEMAND, Conjuntos de datos y esquemas de ECOMMERCE o Esquemas y conjuntos de datos personalizados. En el siguiente ejemplo se muestra cómo añadir un tipo nulo para un campo GENDER.

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
}
```

Conjuntos de datos y esquemas VIDEO_ON_DEMAND

Al crear un grupo de conjuntos de datos de dominio para el dominio VIDEO_ON_DEMAND, cada tipo de conjunto de datos tiene un esquema predeterminado con un conjunto de campos obligatorios y recomendados específicos de VIDEO_ON_DEMAND. Puede utilizar el esquema predeterminado o crear uno nuevo en función del esquema por defecto. Los datos que importe deben coincidir con su esquema en cuanto a formato y tipo. Utilice los esquemas de dominio predeterminados que se enumeran en las secciones siguientes como guía para determinar qué datos debe importar para crear su generador de recomendaciones basado en VIDEO_ON_DEMAND.

Tiene la libertad de añadir campos adicionales. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos

JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Los siguientes temas proporcionan información sobre los campos obligatorios y recomendados de cada conjunto de datos para el dominio VIDEO_ON_DEMAND. Cada sección del conjunto de datos incluye el esquema VIDEO_ON_DEMAND predeterminado en formato JSON.

Temas

- Requisitos de esquema y conjunto de datos de dominio VIDEO_ON_DEMAND
- Requisitos del conjunto de datos de interacciones de elementos (dominio VIDEO_ON_DEMAND)
- Requisitos del conjunto de datos de usuarios (dominio VIDEO_ON_DEMAND)
- Requisitos de conjuntos de datos de elementos (dominio VIDEO_ON_DEMAND)

Requisitos de esquema y conjunto de datos de dominio VIDEO_ON_DEMAND

Cada tipo de conjunto de datos tiene los siguientes campos obligatorios y palabras clave reservadas. Las palabras clave reservadas son campos opcionales que no son de metadatos. Estos campos se consideran reservados porque debe definirlos como su tipo de datos obligatorio cuando los utilice. Los campos de cadenas categóricas reservadas deben tener `categorical` establecido en `true`, mientras que los campos de cadenas reservadas no pueden ser categóricos. Las palabras clave no pueden estar en sus datos.

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
Interacciones de elementos (esquema predeterminado)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`) EVENT_TYPE (`string` y según el caso de uso, los tipos de eventos `Watch` y `Click`)	EVENT_VALUE (`float`, `null`) IMPRESSION (`string`, `null`) RECOMMENDATION_ID (`string`, `null`) EVENT_ATTRIBUTION_SOURCE (`string`, `null`)

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
Usuarios (esquema predeterminado)	USER_ID (`string`) 1 campo de metadatos (`string` categórica o numérica)	SUBSCRIPTION_MODEL (`string` categórica, `null`)
Elementos (esquema predeterminado)	ITEM_ID (`string`) CREATION_TIMESTAMP (`long`) GENRES (`string` categórica)	PRICE (`float`, `null`) DURATION (`float`, `null`) GENRE_L2 (`string` categórica, `null`) GENRE_L3 (`string` categórica, `null`) AVERAGE_RATING (`float`, `null`) PRODUCT_DESCRIPTION (`string` textual, `null`) CONTENT_OWNER (`string` categórica, `null`) CONTENT_CLASSIFICATION (`string` categórica, `null`)

Requisitos del conjunto de datos de interacciones de elementos (dominio VIDEO_ON_DEMAND)

Un conjunto de datos de interacciones de elementos almacena datos históricos y en tiempo real de las interacciones entre los usuarios y los elementos de su catálogo de VIDEO_ON_DEMAND. Para obtener más información acerca de los tipos de datos que puede almacenar en un conjunto de datos de interacciones, consulte Datos de interacciones de elementos.

Debe tener un conjunto de datos de interacciones de elementos para todos los casos de uso y su esquema debe tener los siguientes campos:

- USER_ID (`string`)
- `string` ITEM_ID
- TIMESTAMP (`long`)
- EVENT_TYPE (`string` y según el caso de uso, los tipos de eventos `Watch` y `Click`)

El esquema también puede incluir las siguientes palabras clave reservadas:

- EVENT_VALUE (`float`, `null`)
- IMPRESSION (`string`, `null`)
- RECOMMENDATION_ID (`string`, `null`)

Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos. Para ver un ejemplo de esquema predeterminado de los conjuntos de datos de interacciones de elementos para dominios VIDEO_ON_DEMAND, consulte Esquema de interacciones predeterminado (dominio VIDEO_ON_DEMAND).

Si lo desea, añada la palabra clave reservada EVENT_VALUE si tiene datos de valor para los eventos, como el porcentaje de visualización de un vídeo. Si desea incluir datos de impresiones explícitos e implícitos, también puede añadir la palabra clave reservada IMPRESSION. Para obtener más información acerca del registro de datos de impresiones, consulte Datos de impresiones.

El número máximo total de campos de metadatos opcionales que puede agregar a un conjunto de datos de interacciones de elementos, combinado con el número total de tipos de eventos distintos en sus datos de interacciones de elementos, es de 10. Los campos de metadatos incluidos en este recuento son los campos EVENT_TYPE y EVENT_VALUE, junto con cualquier campo de metadatos personalizados que añada al esquema. El número máximo de campos de metadatos, excluidos los campos reservados, como IMPRESSION, es 5. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene una interacción con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos de un conjunto de datos de interacciones de elementos para el dominio VIDEO_ON_DEMAND, consulte

Cuotas de servicio. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Esquema de interacciones predeterminado (dominio VIDEO_ON_DEMAND)

El siguiente es el esquema de dominio VIDEO_ON_DEMAND predeterminado para los conjuntos de datos de interacciones de elementos.

```
{

  "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"
}
```

Requisitos del conjunto de datos de usuarios (dominio VIDEO_ON_DEMAND)

Un conjunto de datos de usuarios almacena metadatos sobre sus usuarios. Esto puede incluir información como la edad, los intereses, el sexo y el grado de fidelización de cada usuario.

Para obtener información sobre los tipos de datos de usuario que puede importar en Amazon Personalize, consulte Metadatos de usuario. Para obtener información sobre los requisitos de esquema de Amazon Personalize generales, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Un conjunto de datos de usuarios es opcional para todos los casos de uso de VIDEO_ON_DEMAND. Si tiene datos de usuario, te recomendamos crear uno para obtener las recomendaciones más relevantes. Si crea un conjunto de datos de usuarios, su esquema debe incluir los siguientes campos.

- USER_ID
- 1 campo de metadatos (`string` categórica o numérica)

Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos. Para ver un ejemplo del esquema predeterminado de los conjuntos de datos de usuarios para los dominios VIDEO_ON_DEMAND, consulte Esquema de usuarios predeterminado (dominio VIDEO_ON_DEMAND).

Se incluye un campo `SUBSCRIPTION_MODEL` en el esquema predeterminado. Este campo es una palabra clave reservada opcional y debe tener un tipo de `string` con el valor categórico establecido en `true`. Para obtener las mejores recomendaciones, le recomendamos que mantenga este campo en su esquema si tiene información sobre el modelo de suscripción acerca de cada uno de sus usuarios en sus datos. Los datos que importe deben coincidir con su esquema.

Uso de datos categóricos

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en el archivo CSV masivo y en las importaciones de registros individuales. Para los usuarios con varias categorías, separe cada valor con la barra vertical, “|”. Por ejemplo, para un campo SUBSCRIPTION_MODEL, sus datos para un usuario pueden ser estudiante|mensual|descuento.

Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un usuario con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Esquema de usuarios predeterminado (dominio VIDEO_ON_DEMAND)

El siguiente es el esquema de dominio VIDEO_ON_DEMAND predeterminado para conjuntos de datos de usuarios.

```
{

  "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"
}
```

Requisitos de conjuntos de datos de elementos (dominio VIDEO_ON_DEMAND)

Un conjunto de datos de elementos almacena metadatos acerca de los elementos de su catálogo. Esto puede incluir información como el precio, el género y la disponibilidad de cada elemento. Para obtener información sobre los tipos de datos de elementos que puede importar en Amazon Personalize, consulte Metadatos de elemento. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Para algunos casos de uso, se requiere un conjunto de datos de elementos (consulte Casos de uso de VIDEO_ON_DEMAND). Si es opcional, recomendamos crear uno para obtener las recomendaciones más relevantes. Si crea un conjunto de datos de elementos, su esquema debe incluir los siguientes campos:

- ITEM_ID
- GENRES (`string` categórica)
- CREATION_TIMESTAMP (en formato de tiempo Unix)

El esquema también puede incluir las siguientes palabras clave reservadas. Cada palabra clave indica su tipo de datos obligatorio y indica si admite datos nulos. La adición del tipo nulo es opcional.

- PRICE (float)
- DURATION (float)
- GENERE_L2 (`string` categórica, `null`)
- GENRE_L3 (`string` categórica, `null`)
- AVERAGE_RATING (`float`, `null`)
- PRODUCT_DESCRIPTION (`string` textual, `null`)
- CONTENT_OWNER (`string` categórica, `null`): la empresa propietaria del vídeo. Por ejemplo, los valores pueden ser HBO, Paramount y NBC.
- CONTENT_CLASSIFICATION (`string` categórica, `null`): la clasificación del contenido. Por ejemplo, los valores pueden ser G, PG, PG-13, R, NC-17 y no sin clasificación.

Para obtener las mejores recomendaciones, le recomendamos que mantenga tantos campos de estos en el esquema como datos tenga. Los datos que importe deben coincidir con su esquema. El número máximo de columnas de metadatos es 100. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Utilice las palabras clave reservadas GENRE_L2 y GENRE_L3 para los elementos con varias categorías de múltiples niveles. Para obtener más información, consulte Uso de datos categóricos. Para obtener información sobre metadatos textuales y categóricos, consulte Preparación de los metadatos de elemento para el entrenamiento. Para ver un ejemplo del esquema predeterminado para los conjuntos de datos de elementos para dominios de ECOMMERCE, consulte Esquema de elementos predeterminado (dominio VIDEO_ON_DEMAND).

Uso de datos categóricos

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en su archivo CSV masivo y en las importaciones de elementos individuales. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un elemento con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para los elementos con varias categorías, separe cada valor con la barra vertical, "|". Por ejemplo, para un campo GENRES, los datos para un elemento podrían ser `Action|Crime|Biopic`. Si tiene varios niveles de datos categóricos y algunos elementos tienen varias categorías para cada nivel de la jerarquía, agregue un campo para cada nivel y un indicador de nivel después del nombre de cada campo: GENRES, GENRE_L2, GENRE_L3. Esto le permite filtrar las recomendaciones en función de las subcategorías, incluso si un elemento pertenece a varias categorías de múltiples niveles. Por ejemplo, un vídeo puede tener los siguientes datos para cada nivel de categoría:

- GENRES: Acción|Aventura
- GENRE_L2: Crimen|Western
- GENRE_L3: biopic

En este ejemplo, el vídeo está en la jerarquía acción > crimen > biopic y la jerarquía aventura > western > biopic. Recomendamos usar solo hasta L3, pero puede usar más niveles si es necesario. Para obtener información sobre la creación y el uso de filtros, consulte Recomendaciones de filtrado y segmentos de usuarios.

Esquema de elementos predeterminado (dominio VIDEO_ON_DEMAND)

El siguiente es el esquema predeterminado para los conjuntos de datos de elementos del dominio VIDEO_ON_DEMAND.

```
{
  "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"
}
```

Conjuntos de datos y esquemas de ECOMMERCE

Al crear un grupo de conjuntos de datos de dominio para el dominio de ECOMMERCE, cada tipo de conjunto de datos tiene un esquema predeterminado con un conjunto de campos obligatorios y recomendados específicos de ECOMMERCE. Puede usar el esquema predeterminado o crear uno nuevo basado en el esquema predeterminado. Los datos que importe deben coincidir con su esquema en cuanto a formato y tipo. Use los esquemas de dominio predeterminados que se enumeran en las secciones siguientes como guía para determinar qué datos importar para crear su generador de recomendaciones basado en ECOMMERCE.

Tiene la libertad de añadir campos adicionales. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Los siguientes temas proporcionan información sobre los campos obligatorios y recomendados de cada conjunto de datos para el dominio de ECOMMERCE. Cada sección del conjunto de datos incluye el esquema de ECOMMERCE predeterminado en formato JSON.

Temas

- Requisitos de esquema y conjunto de datos del dominio de ECOMMERCE

- Requisitos del conjunto de datos de interacciones de elementos (dominio ECOMMERCE)
- Requisitos del conjunto de datos de usuarios (dominio de ECOMMERCE)
- Requisitos del conjunto de datos de elementos (dominio de ECOMMERCE)

Requisitos de esquema y conjunto de datos del dominio de ECOMMERCE

Cada tipo de conjunto de datos tiene los siguientes campos obligatorios y palabras clave reservadas. Las palabras clave reservadas son campos opcionales que no son de metadatos. Estos campos se consideran reservados porque debe definirlos como su tipo de datos obligatorio cuando los utilice. Los campos de cadenas categóricas reservadas deben tener `categorical` establecido en `true`, mientras que los campos de cadenas reservadas no pueden ser categóricos. Las palabras clave no pueden estar en sus datos.

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
Interacciones de elementos (esquema predeterminado)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`) EVENT_TYPE (`string` y según el caso de uso, los tipos de eventos `Purchase` y `View`)	EVENT_VALUE (`float`, `null`) IMPRESSION (`string`, `null`) RECOMMENDATION_ID (`string`, `null`) EVENT_ATTRIBUTION_SOURCE (`string`, `null`)
Usuarios (esquema predeterminado)	USER_ID (`string`) 1 campo de metadatos (`string` categórica o numérica)	
Elementos (esquema predeterminado)	ITEM_ID (`string`) PRICE (`float`)	CATEGORY_L2 (`string`, categórica, `null`) CATEGORY_L3 (`string`, categórica, `null`)

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
	CATEGORY_L1 (`string` categórica)	PRODUCT_DESCRIPTION (`string` textual, `null`) CREATION_TIMESTAMP (`long`) AGE_GROUP (`string` categórica, `null`) ADULT (`string` categórica, `null`) GENDER (`string` categórica, `null`)

Requisitos del conjunto de datos de interacciones de elementos (dominio ECOMMERCE)

Un conjunto de datos de interacciones de elementos almacena datos históricos y en tiempo real de las interacciones entre los usuarios y los elementos de su catálogo de ECOMMERCE. Para obtener más información acerca de los tipos de datos que puede almacenar en un conjunto de datos de interacciones, consulte Datos de interacciones de elementos. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Como mínimo, debe crear un conjunto de datos de interacciones de elementos y su esquema debe tener los siguientes campos:

- USER_ID (`string`)
- ITEM_ID (`string`)
- TIMESTAMP (`long`)
- EVENT_TYPE (`string` y según el caso de uso, los tipos de eventos `Purchase` y `View`)

El esquema también puede incluir las siguientes palabras clave reservadas:

- EVENT_VALUE (`float`, `null`)
- IMPRESSION (`string`, `null`)
- RECOMMENDATION_ID (`string`, `null`)

Los datos que importe deben coincidir con su esquema. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos. Para ver un ejemplo del esquema predeterminado de los conjuntos de datos de interacciones de elementos para dominios ECOMMERCE, consulte Esquema de interacciones predeterminado (dominio de ECOMMERCE).

Si lo desea, añada la palabra clave reservada EVENT_VALUE si tiene datos de valor para eventos. Si desea incluir datos de impresiones explícitos e implícitos, también puede añadir la palabra clave reservada IMPRESSION. Para obtener más información acerca del registro de datos de impresiones, consulte Datos de impresiones.

El número máximo total de campos de metadatos opcionales que puede agregar a un conjunto de datos de interacciones de elementos, combinado con el número total de tipos de eventos distintos en sus datos de interacciones de elementos, es de 10. Los campos de metadatos incluidos en este recuento son los campos EVENT_TYPE y EVENT_VALUE, junto con cualquier campo de metadatos personalizados que añada al esquema. El número máximo de campos de metadatos, excluidos los campos reservados, como IMPRESSION, es 5. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene una interacción con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos de un conjunto de datos de interacciones de elementos para el dominio ECOMMERCE, consulte Cuotas de servicio.

Esquema de interacciones predeterminado (dominio de ECOMMERCE)

El siguiente es el esquema del dominio ECOMMERCE predeterminado para los conjuntos de datos de interacciones de elementos.

```
{
```

```
  "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"
}
```

Requisitos del conjunto de datos de usuarios (dominio de ECOMMERCE)

Un conjunto de datos de usuarios almacena metadatos sobre sus usuarios. Esto puede incluir información como la edad, el sexo y el grado de fidelización para cada usuario. Para obtener más información sobre los tipos de datos de usuario que puede importar en Amazon Personalize, consulte Metadatos de usuario. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Un conjunto de datos de usuarios es opcional para todos los casos de uso de ECOMMERCE. Si tiene datos de usuario, te recomendamos crear uno para obtener las recomendaciones más relevantes. Si crea un conjunto de datos de usuarios, su esquema debe incluir los siguientes campos.

- USER_ID
- 1 campo de metadatos (`string` categórica o numérica)

Los datos que importe deben coincidir con su esquema. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos. Para ver un ejemplo del esquema predeterminado de los conjuntos de datos de usuarios para dominios de ECOMMERCE, consulte Esquema de usuarios predeterminado (dominio de ECOMMERCE).

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de usuarios, consulte Cuotas de servicio.

Uso de datos categóricos

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en el archivo CSV masivo y en las importaciones de registros individuales. Para los usuarios con varias categorías, separe cada valor con la barra vertical, "|". Por ejemplo, para un campo SUBSCRIPTION_MODEL, sus datos para un usuario pueden ser estudiante|mensual|descuento.

Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un usuario con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Esquema de usuarios predeterminado (dominio de ECOMMERCE)

El siguiente es el esquema de dominio de ECOMMERCE predeterminado para los conjuntos de datos de usuarios con un campo CATEGORY como el campo de metadatos obligatorio.

```
{

  "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"
}
```

Requisitos del conjunto de datos de elementos (dominio de ECOMMERCE)

Un conjunto de datos de elementos almacena metadatos sobre sus elementos de ECOMMERCE. Esto puede incluir información como el precio, la categoría y la descripción del producto de cada elemento. Para obtener más información sobre los tipos de datos de elementos que puede importar en Amazon Personalize, consulte Metadatos de elemento. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas, con independencia del dominio.

Un conjunto de datos de elementos es opcional para todos los casos de uso de ECOMMERCE. Si tiene datos de elementos, recomendamos crear uno para obtener las recomendaciones más relevantes. Si crea un conjunto de datos de elementos, su esquema debe incluir los siguientes campos:

- ITEM_ID
- PRICE (`float`)
- CATEGORY_L1 (`string` categórica)

El esquema también puede incluir las siguientes palabras clave reservadas. Para los campos categóricos, puede definir su propio rango de valores en función de su caso de uso.

- CATEGORY_L2 (`string`, categórica, `null`)
- CATEGORY_L3 (`string`, categórica, `null`)
- PRODUCT_DESCRIPTION (`string` textual, `null`)
- CREATION_TIMESTAMP (`float`)
- AGE_GROUP (`string` categórica, `null`): el grupo de edad al que pertenece el elemento. Los valores pueden ser recién nacidos, bebés, niños y adultos.
- ADULTO (`string` categórica, `null`): si el elemento está restringido solo a adultos, por ejemplo, bebidas alcohólicas. Los valores pueden ser sí o no.

- GÉNERO (`string` categórica, `null`): el género al que corresponde el elemento. Los valores pueden ser hombre, mujer y unisex.

Para obtener las mejores recomendaciones, le recomendamos que mantenga tantos campos de estos en el esquema como datos tenga. Los datos que importe deben coincidir con su esquema. Los datos que importe deben coincidir con su esquema. El número máximo de columnas de metadatos es 100. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Utilice las palabras clave reservadas CATEGORY_L2 y CATEGORY_L3 para los elementos con varias categorías de múltiples niveles. Para obtener más información, consulte Uso de datos categóricos. Para obtener información sobre metadatos textuales y categóricos, consulte Metadatos de texto no estructurado. Para ver un ejemplo del esquema predeterminado para los conjuntos de datos de elementos para dominios de ECOMMERCE, consulte Esquema de elementos predeterminados (dominio de ECOMMERCE).

Uso de datos categóricos

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en su archivo CSV masivo y en las importaciones de elementos individuales. Puede definir su propio rango de valores en función de su caso de uso. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un elemento con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para los elementos con varias categorías, separe cada valor con la barra vertical, "|". Por ejemplo, para un campo CATEGORY_L1, los datos para un elemento podrían ser `Electronics|Productivity|Mouse`. Si tiene varios niveles de datos categóricos y algunos elementos tienen varias categorías para cada nivel de la jerarquía, agregue un campo para cada nivel y un indicador de nivel después del nombre de cada campo: CATEGORY_L1, CATEGORY_L2, CATEGORY_L3. Esto le permite filtrar las recomendaciones en función de las subcategorías, incluso si un elemento pertenece a varias categorías de múltiples niveles. Por ejemplo, un elemento puede tener los siguientes datos para cada nivel de categoría:

- CATEGORY_L1: Electrónica|Productividad
- CATEGORY_L2: Productividad|Ordenadores
- CATEGORY_L3: Ratón

En este ejemplo, el elemento se encuentra en la jerarquía electrónica > productividad > ratón y productividad > ordenadores > ratón. Recomendamos usar solo hasta L3, pero puede usar más niveles si es necesario. Para obtener información sobre la creación y el uso de filtros, consulte Recomendaciones de filtrado y segmentos de usuarios.

Esquema de elementos predeterminados (dominio de ECOMMERCE)

El siguiente es el esquema predeterminado para conjuntos de datos de elementos para el dominio de ECOMMERCE con solo los campos obligatorios.

```
{
  "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"
}
```

Esquemas y conjuntos de datos personalizados

Cuando crea un grupo de conjuntos de datos personalizados, crea sus propios esquemas desde cero. Los conjuntos de datos y esquemas de grupos de conjuntos de datos personalizados tienen menos campos obligatorios y mayor flexibilidad. En los siguientes temas se explican los requisitos de esquema y de datos para los conjuntos de datos de un grupo de conjuntos de datos personalizados.

Cada sección del conjunto de datos enumera los datos necesarios para el tipo de conjunto de datos y proporciona un ejemplo de JSON de un esquema.

Para obtener información sobre los tipos de datos que puede importar en Amazon Personalize, consulte Tipos de datos que Amazon Personalize puede utilizar. Para obtener información sobre los requisitos generales de esquema de Amazon Personalize, como los requisitos de formato y los tipos de datos de campo disponibles, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize. Estos requisitos se aplican a todos los esquemas de Amazon Personalize.

Temas

- Requisitos de esquemas y conjuntos de datos personalizados
- Requisitos del esquema del conjunto de datos de interacciones de elementos (personalizado)
- Requisitos del esquema del conjunto de datos de usuarios (personalizado)
- Requisitos del esquema del conjunto de datos de elementos (personalizado)
- Requisitos del esquema del conjunto de datos de acciones (personalizado)
- Requisitos del esquema del conjunto de datos de interacciones de acción (personalizado)

Requisitos de esquemas y conjuntos de datos personalizados

Al crear un conjunto de datos para un grupo de conjuntos de datos personalizados, cada tipo de conjunto de datos tiene los siguientes campos obligatorios y palabras clave reservadas con los tipos de datos obligatorios.

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
Interacciones de elementos (ejemplo de esquema)	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`)	EVENT_TYPE (`string`) EVENT_VALUE (`float`, `null`) IMPRESSION (`string`, `null`) RECOMMENDATION_ID (`string`, `null`)

Tipo de conjunto de datos	Campos obligatorios	Palabras clave reservadas
		EVENT_ATTRIBUTION_SOURCE (`string`, `null`)
Usuarios (ejemplo de esquema)	USER_ID (`string`) 1 campo de metadatos (`string` categórica o numérica)	
Elementos (ejemplo de esquema)	ITEM_ID (`string`) 1 campo de metadatos (campo `string` categórico o textual o campo numérico)	CREATION_TIMESTAMP (`long`)
Acciones (ejemplo de esquema)	ACTION_ID (`string`) 1 campo de metadatos (`string` categórica o numérica)	CREATION_TIMESTAMP (`long`) VALUE (`long`, `null`) TYPE (`string`, `null`) EXPIRATION_TIMESTAMP (`long`, `null`) REPEAT_FREQUENCY (`long`, `null`)
Interacciones de acción (ejemplo de esquema)	USER_ID (`string`) ACTION_ID (`string`) EVENT_TYPE (`string`) TIMESTAMP (`long`)	IMPRESSION (`string`, `null`) RECOMMENDATION_ID (`string`, `null`)

Campos de metadatos

Los metadatos incluyen campos de cadena o sin cadena que no son obligatorios o que no utilizan una palabra clave reservada. Los esquemas de metadatos tienen las siguientes restricciones:

- Los esquemas de usuarios, elementos y acciones requieren al menos un campo de metadatos.
- Puede agregar como máximo 25 campos de metadatos para un esquema de usuarios, 100 campos de metadatos para un esquema de elementos y 10 campos de metadatos para un esquema de acciones.
- Si agrega su propio campo de metadatos de tipo `string`, debe incluir el atributo `categorical` o el atributo `textual` (solo los esquemas de elementos admiten campos con el atributo textual). De lo contrario, Amazon Personalize no utilizará el campo al entrenar un modelo.

Palabras clave reservadas

Las palabras clave reservadas son campos opcionales que no son de metadatos. Estos campos se consideran reservados porque debe definirlos como su tipo de datos obligatorio al usarlos, y las palabras clave no se pueden usar como valores en sus datos. Los campos de cadenas categóricas reservadas deben tener `categorical` establecido en `true`, mientras que los campos de cadenas reservadas no pueden ser categóricos. Las siguientes son palabras clave reservadas:

- EVENT_TYPE: para los conjuntos de datos de interacciones de elementos con uno o más tipos de eventos, como hacer clic y descargar, use un campo `EVENT_TYPE`. Debe definir un campo EVENT_TYPE como `string` y no puede establecerse como categórico.
- EVENT_VALUE: para los conjuntos de datos de interacciones de elementos que incluyen datos de valor para los eventos, como el porcentaje de un vídeo que ha visto un usuario, use un campo `EVENT_VALUE` con el tipo `float` y, de forma opcional, `null`.
- CREATION_TIMESTAMP: para los conjuntos de datos de elementos o acciones con una marca de tiempo para la fecha de creación de cada elemento, use un campo `CREATION_TIMESTAMP` con un tipo `long`. Amazon Personalize utiliza los datos de `CREATION_TIMESTAMP` para calcular la edad de un elemento y ajustar las recomendaciones en consecuencia. Consulte Datos de la marca de tiempo de creación.
- IMPRESSION: para los conjuntos de datos de interacciones de elementos con datos de impresiones explícitos, utilice un campo `IMPRESSION` con el tipo `String` y, si lo desea, el tipo `null`. Las impresiones son listas de elementos que estaban visibles para un usuario cuando

interactuaba con un elemento concreto (por ejemplo, al hacer clic o verlo). Para obtener más información, consulte Datos de impresiones.

- RECOMMENDATION_ID: para los conjuntos de datos de interacciones de elementos que utilizan recomendaciones anteriores como datos de impresiones implícitas, si lo desea, utilice un campo `RECOMMENDATION_ID` con el tipo `String` y, opcionalmente, el tipo `null`.

  No es necesario añadir un campo `RECOMMENDATION_ID` para que Amazon Personalize utilice impresiones implícitas al generar recomendaciones. Puede pasar un valor de `recommendationId` en una operación PutEvents sin él. Para obtener más información, consulte Datos de impresiones.
- VALUE: para los conjuntos de datos de acciones, si tiene datos de valor para algunas de sus acciones o todas, agregue un campo `VALUE` al esquema. Para el tipo, use `long` y, si lo desea, escriba `null`. Para obtener más información sobre las acciones y su valor, consulte Value data.
- ACTION_EXPIRATION_TIMESTAMP: para los conjuntos de datos de acciones, si tiene una marca de tiempo de caducidad para algunas de sus acciones o todas, agregue un campo `ACTION_EXPIRATION_TIMESTAMP` al esquema. Para el tipo, use `long` y, si lo desea, escriba `null`. Para obtener más información sobre las marcas de tiempo de caducidad, consulte Datos de la marca de tiempo de caducidad de la acción.
- REPEAT_FREQUENCY: para los conjuntos de datos de acciones, si tiene datos de frecuencia de repetición para algunas de sus acciones o todas, agregue un campo `REPEAT_FREQUENCY` al esquema. Para el tipo, use `long` y, si lo desea, escriba `null`. Para obtener más información sobre los datos de frecuencia de repetición, consulte Datos de frecuencia de repetición.

Requisitos del esquema del conjunto de datos de interacciones de elementos (personalizado)

Un conjunto de datos de interacciones de elementos almacena datos históricos y en tiempo real de las interacciones entre los usuarios y los elementos de su catálogo. Para obtener información sobre los tipos de datos de interacciones que Amazon Personalize puede utilizar, consulte Datos de interacciones de elementos.

Los datos que proporcione para cada interacción deben coincidir con su esquema. Según su esquema, los metadatos de interacción pueden incluir valores nulos o vacíos. Como mínimo, debe proporcionar lo siguiente para cada interacción:

- ID de usuario

- ID de elemento
- Marca temporal (en formato de tiempo Unix)

Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

El número máximo total de campos de metadatos opcionales que puede agregar a un conjunto de datos de interacciones de elementos, combinado con el número total de tipos de eventos distintos en sus datos de interacciones de elementos, es de 10. Los campos de metadatos incluidos en este recuento son los campos EVENT_TYPE y EVENT_VALUE, junto con cualquier campo de metadatos personalizados que añada al esquema. El número máximo de campos de metadatos, excluidos los campos reservados, como IMPRESSION, es 5. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene una interacción con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de interacciones de elementos, consulte Cuotas de servicio.

Ejemplo de esquema de interacciones (personalizado)

En el siguiente ejemplo, se muestra un esquema para un conjunto de datos de interacciones de elementos. Los campos `USER_ID`, `ITEM_ID` y `TIMESTAMP` son obligatorios. Los campos `EVENT_TYPE`, `EVENT_VALUE` y `IMPRESSION` son palabras clave reservadas reconocidas por Amazon Personalize. EVENT_TYPE debe ser de tipo cadena y no puede ser categórico. `LOCATION` y `DEVICE` son campos de metadatos contextuales opcionales. Para obtener información sobre los requisitos de esquema, consulte Requisitos de esquemas y conjuntos de datos personalizados.

```
{

  "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"
}
```

Para este esquema, las primeras líneas de datos históricos de un archivo CSV pueden tener el siguiente aspecto. Tenga en cuenta que algunos valores para EVENT_VALUE son nulos.

```
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
...
...
```

Requisitos del esquema del conjunto de datos de usuarios (personalizado)

Un conjunto de datos de usuarios almacena metadatos sobre sus usuarios. Esto puede incluir información como la edad, el sexo y el grado de fidelización de cada elemento. Para obtener información sobre los tipos de datos de usuario que puede importar en Amazon Personalize, consulte Metadatos de usuario.

Los datos que proporcione para cada usuario deben coincidir con su esquema. Como mínimo, debe proporcionar un ID para cada usuario (longitud máxima de 256 caracteres). Según el esquema, los metadatos del usuario pueden incluir empty/null valores. El esquema de usuarios debe tener como mínimo un campo de metadatos, pero si agrega un tipo `null`, este valor puede ser nulo para el usuario. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en el archivo CSV masivo y en las importaciones de registros individuales. Para los usuarios con varias categorías, separe cada valor con la barra vertical, “|”. Por ejemplo, para un campo SUBSCRIPTION_MODEL, sus datos para un usuario pueden ser estudiante|mensual|descuento.

Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un usuario con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de usuarios, consulte Cuotas de servicio.

Ejemplo de esquema de usuarios (personalizado)

En el siguiente ejemplo se muestra cómo estructurar un esquema de usuarios. El campo `USER_ID` es obligatorio y los campos `AGE` y `GENDER` son metadatos. Se requiere al menos un campo de metadatos y puede añadir como máximo 25 campos de metadatos. Para obtener información sobre los requisitos de esquema, consulte Requisitos de esquemas y conjuntos de datos personalizados.

```
{
  "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"
}
```

Para este esquema, las primeras líneas de datos históricos de un archivo CSV pueden tener el siguiente aspecto.

```
USER_ID,AGE,GENDER
5,34,Male
6,56,Female
8,65,Male
...
...
```

Requisitos del esquema del conjunto de datos de elementos (personalizado)

Un conjunto de datos de elementos almacena metadatos acerca de los elementos de su catálogo. Esto puede incluir información como el precio, el género y la disponibilidad de cada elemento. Para obtener información sobre los tipos de datos de elementos que puede importar en Amazon Personalize, consulte Metadatos de elemento.

Los datos que proporciones para cada elemento deben coincidir con su esquema de conjunto de datos de elementos. Como mínimo, debe proporcionar un ID de elemento para cada usuario (longitud máxima de 256 caracteres). Según el esquema, los metadatos de los elementos pueden incluir empty/null valores. El esquema debe tener como mínimo un campo de metadatos, pero si agrega un tipo `null`, este valor puede ser nulo para el elemento. Tiene la libertad de añadir campos adicionales según su caso de uso y sus datos. Siempre que los campos no figuren como obligatorios o reservados, y los tipos de datos figuren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para usar datos categóricos, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en su esquema. A continuación, incluya los datos categóricos en su archivo CSV masivo y en las importaciones de elementos individuales. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene un elemento con un valor categórico con más de 1000 caracteres, se generará un error en su trabajo de importación del conjunto de datos.

Para los elementos con varias categorías, separe cada valor con la barra vertical, “|”. Por ejemplo, para un campo GENRES, los datos para un elemento podrían ser `Action|Crime|Biopic`. Si tiene varios niveles de datos categóricos y algunos elementos tienen varias categorías para cada nivel de la jerarquía, agregue un campo para cada nivel y un indicador de nivel después del nombre de cada campo: GENRES, GENRE_L2, GENRE_L3. Esto le permite filtrar las recomendaciones en función de las subcategorías, incluso si un elemento pertenece a varias categorías de múltiples niveles (para obtener información sobre la creación y el uso de filtros, consulte Recomendaciones de filtrado y segmentos de usuarios). Por ejemplo, un vídeo puede tener los siguientes datos para cada nivel de categoría:

- GENRES: Acción|Aventura
- GENRE_L2: Crimen|Western
- GENRE_L3: Biopic

En este ejemplo, el vídeo está en la jerarquía acción > crimen > biopic y la jerarquía aventura > western > biopic. Recomendamos usar solo hasta L3, pero puede usar más niveles si es necesario.

Durante el entrenamiento de modelos, Amazon Personalize considera un máximo de 750 000 elementos. Si importa más de 750 000 elementos, Amazon Personalize decide qué elementos incluir en el entrenamiento, haciendo hincapié en incluir elementos nuevos (elementos que ha añadido recientemente sin interacciones) y elementos existentes con datos de interacciones recientes.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de elementos, consulte Cuotas de servicio.

Ejemplo de esquema de conjunto de datos de elementos (personalizado)

En el siguiente ejemplo se muestra cómo estructurar un esquema de elementos. El campo `ITEM_ID` es obligatorio. El campo `GENRE` son metadatos categóricos y el campo `DESCRIPTION` son metadatos textuales. Se requiere un campo de metadatos como mínimo. Puede agregar un máximo de 100 campos de metadatos. El campo `CREATION_TIMESTAMP` es una palabra clave reservada. Para obtener información sobre los requisitos de esquema, consulte Requisitos de esquemas y conjuntos de datos personalizados.

```
{
  "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"
}
```

Para este esquema, las primeras líneas de datos históricos de un archivo CSV pueden tener el siguiente aspecto.

```
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."
...
...
```

Requisitos del esquema del conjunto de datos de acciones (personalizado)

Una acción es una actividad de interacción que puede que quiera recomendar a sus clientes. Entre las acciones puede incluirse instalar su aplicación móvil, completar un perfil de pertenencia a grupo, unirse a un programa de fidelización o registrarse para recibir correos electrónicos promocionales. Un conjunto de datos de acciones almacena datos sobre las acciones. Para obtener información sobre los tipos de datos de acciones que puede importar en Amazon Personalize, consulte Metadatos de acción.

Los datos que proporcione para cada acción deben coincidir con el esquema del conjunto de datos de acciones. Según el esquema, los metadatos de las acciones pueden incluir empty/null valores.

Como mínimo, debe proporcionar un ID de acción para cada usuario (longitud máxima de 256 caracteres). El esquema debe tener como mínimo un campo de metadatos, pero si agrega un

tipo `null`, este valor puede ser nulo para la acción. Puede agregar campos adicionales en función de su caso de uso y sus datos. A menos que los campos figuren como obligatorios o reservados y los tipos de datos se enumeren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para agregar un campo categórico, agregue un campo de tipo `string` y establezca el atributo categórico del campo en `true` en el esquema. A continuación, incluya los datos categóricos en su archivo CSV masivo y en las importaciones de acciones individuales. Los valores categóricos pueden tener 1000 caracteres como máximo. Si tiene una acción con un valor categórico con más de 1000 caracteres, se generará un error en el trabajo de importación del conjunto de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de acciones, consulte Cuotas de servicio.

Ejemplo de esquema de conjunto de datos de acciones (personalizado)

En el siguiente ejemplo se muestra cómo estructurar un esquema de acciones. El campo `ACTION_ID` es obligatorio. `MEMBERSHIP_LEVEL` es un campo de cadena categórico. Los campos `VALUE`, `CREATION_TIMESTAMP` y `REPEAT_FREQUENCY` son palabras clave reservadas con los tipos necesarios. Puede agregar un máximo de 10 columnas. Para obtener información sobre los requisitos de esquema, consulte Requisitos de esquemas y conjuntos de datos personalizados.

```
{
  "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"
}
```

Para este esquema, las primeras líneas de datos históricos de un archivo CSV pueden tener el siguiente aspecto.

```
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
...
...
```

Requisitos del esquema del conjunto de datos de interacciones de acción (personalizado)

Un conjunto de datos de interacciones de acción almacena datos históricos y en tiempo real de las interacciones entre los usuarios y las acciones en su conjunto de datos de acciones. Para obtener información sobre los tipos de datos que Amazon Personalize puede utilizar, consulte Datos de interacciones de acción.

Los datos que proporcione para cada interacción deben coincidir con su esquema. Según su esquema, los metadatos de interacción pueden incluir valores nulos o vacíos. Como mínimo, el esquema debe incluir lo siguiente:

- USER_ID
- ACTION_ID
- TIMESTAMP
- EVENT_TYPE

Puede agregar campos adicionales en función de su caso de uso y sus datos. A menos que los campos figuren como obligatorios o reservados y los tipos de datos se enumeren en Tipos de datos de esquema, puede elegir los nombres de los campos y los tipos de datos.

Para obtener más información sobre los requisitos mínimos y los límites máximos de datos para un conjunto de datos de interacciones de acción, consulte Cuotas de servicio.

Ejemplo de esquema de conjunto de datos de interacciones de acción (personalizado)

En el siguiente ejemplo, se muestra un esquema para un conjunto de datos de interacciones de acción solo con los campos necesarios. Para obtener información sobre los requisitos generales de formato del esquema, consulte Requisitos de formato de esquema.

```
{

  "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"
}
```

Para este esquema, las primeras líneas de datos históricos de un archivo CSV pueden tener el siguiente aspecto.

```
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
...
...
```

Lista de verificación de preparación

Esta lista de verificación proporciona listas de las características, los requisitos y la orientación sobre datos de Amazon Personalize. Puede ayudarle en sus tareas de planificación, aunque también puede utilizarla como referencia para crear recursos en Amazon Personalize.

Temas

- ¿Ha adaptado sus casos de uso a los recursos de Amazon Personalize?
- ¿Tiene suficientes datos de interacciones de elementos?
- ¿Cuenta con una arquitectura de streaming de eventos en tiempo real?
- ¿Están sus datos optimizados para Amazon Personalize?
- ¿Recopila datos opcionales que puedan mejorar las recomendaciones?
- ¿Tiene un plan para poner a prueba sus recomendaciones?
- ¿Tiene objetivos empresariales adicionales?

¿Ha adaptado sus casos de uso a los recursos de Amazon Personalize?

Las recomendaciones de Amazon Personalize pueden abordar los siguientes casos de uso:

- Generar recomendaciones personalizadas para un usuario
- Recomendar elementos similares o relacionados
- Recomendar elementos populares o de moda
- Recomendación de las siguientes mejores acciones para un usuario
- Reordenar por relevancia (solo con recursos personalizados)
- Generar segmentos de usuarios (solo con recursos personalizados)

Amazon Personalize incluye recursos basados en dominios y recursos personalizados que se han configurado para estos casos de uso. Comience por crear un grupo de conjuntos de datos de dominio o un grupo de conjuntos de datos personalizados:

- Con un grupo de conjuntos de datos de dominio, creará recursos preconfigurados y optimizados para los dominios VIDEO_ON_DEMAND o ECOMMERCE.

Si tiene una aplicación de streaming de vídeo o de comercio electrónico, le recomendamos que comience con un grupo de conjuntos de datos de dominio. Aún puede añadir recursos personalizados, como soluciones y versiones de soluciones entrenadas para casos de uso personalizados. Además, puede seguir usando recursos personalizados para obtener recomendaciones por lotes. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio.

- Con un grupo de conjuntos de datos personalizados, puede elegir una receta que se adapte a su caso de uso. A continuación, entrenará e implementará únicamente soluciones y versiones de soluciones configurables (modelos de recomendación entrenados de Amazon Personalize). Cuando esté listo, podrá implementar la versión de solución en una campaña para obtener recomendaciones en tiempo real. O puede obtener recomendaciones por lotes sin una campaña.

  Si tiene una aplicación de streaming de vídeo o de comercio electrónico, le recomendamos que cree con un grupo de conjuntos de datos personalizado. De lo contrario, comience con un grupo de conjuntos de datos de dominio y agregue recursos personalizados según sea necesario.

Para obtener información sobre los casos de uso y las recetas personalizadas disponibles en Amazon Personalize, consulte Correlación del caso de uso con los recursos de Amazon Personalize.

¿Tiene suficientes datos de interacciones de elementos?

Para todos los casos de uso y recetas, debe tener un mínimo de 1000 interacciones de elementos para 25 usuarios únicos con al menos dos interacciones cada uno. Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

Si no está seguro de tener suficientes datos, puede importarlos y analizarlos con la consola Amazon Personalize. Para obtener más información, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

¿Cuenta con una arquitectura de streaming de eventos en tiempo real?

Si no tiene suficientes datos de interacciones de elementos, puede utilizar Amazon Personalize para recopilar datos de eventos adicionales en tiempo real. Con algunas recetas y casos de uso,

Amazon Personalize puede aprender de la actividad más reciente de sus usuarios y actualizar las recomendaciones a medida que utilizan su aplicación.

Para obtener información sobre el registro de eventos, incluida la forma en que los eventos afectan a las recomendaciones, una lista de servicios de seguimiento de eventos de terceros y ejemplos de implementaciones, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.

¿Están sus datos optimizados para Amazon Personalize?

Le recomendamos que compruebe lo siguiente en sus datos:

- Compruebe si faltan valores. Recomendamos que un mínimo del 70 % de sus registros contenga datos para cada atributo. Recomendamos que las columnas que permiten valores nulos estén completas al menos en un 70 %.
- Corrija cualquier imprecisión o problema en los datos, como las convenciones de nomenclatura incoherentes, las categorías duplicadas de un elemento, los identificadores no coincidentes entre conjuntos de datos o los identificadores duplicados. Estos problemas pueden afectar negativamente a las recomendaciones o generar un comportamiento inesperado. Por ejemplo, es posible que sus datos incluyan tanto “N/A” como “No aplicable”, pero que filtre las recomendaciones basándose únicamente en “N/A”. El filtro no tendría en cuenta los elementos marcados como “No aplicable”.
- Si un elemento, usuario o acción puede tener varias categorías, como una película con varios géneros, combine los valores categóricos en un atributo y separe cada valor con el operador |. Por ejemplo, los datos de GÉNEROS de una película pueden ser Acción | Aventura | Thriller.
- Evite tener más de 1000 categorías posibles para una columna (a menos que la columna contenga datos únicamente con fines de filtrado).

Para obtener una lista completa de recomendaciones de datos e instrucciones sobre cómo puede usar Amazon Personalize para identificar problemas, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

¿Recopila datos opcionales que puedan mejorar las recomendaciones?

Los siguientes datos pueden ayudar a mejorar la pertinencia de sus recomendaciones.

- Tipo de evento (obligatorio para todos los casos de uso de grupos de conjuntos de datos de dominio)
- Valor del evento
- Metadatos contextuales
- Metadatos de elementos y usuarios
- Datos de interacción de acciones (que solo utilizan las recetas PERSONALIZED_ACTIONS)

Para obtener más información sobre los tipos de datos que Amazon Personalize puede utilizar, consulte Tipos de datos que Amazon Personalize puede utilizar.

¿Tiene un plan para poner a prueba sus recomendaciones?

Puede usar las pruebas A/B para comparar los resultados de diferentes grupos de usuarios que interactúan con recomendaciones de diferentes modelos. Las pruebas A/B pueden ayudarle a comparar diferentes estrategias de recomendación y comprobar si las recomendaciones le ayudan a alcanzar sus objetivos empresariales. Para obtener más información, consulte Medir el impacto de las recomendaciones mediante A/B pruebas.

¿Tiene objetivos empresariales adicionales?

En algunos casos es posible que tenga otros objetivos además de la generación de recomendaciones pertinentes para sus usuarios. Por ejemplo, es posible que desee maximizar los ingresos o promocionar determinados tipos de elementos de una categoría determinada. Las siguientes funciones de Amazon Personalize pueden servirle de ayuda:

- Promociones: puede utilizar las promociones para asegurarse de que un determinado porcentaje de elementos satisfaga los requisitos de su empresa. Para obtener más información, consulte Promoción de elementos en las recomendaciones en tiempo real.
- Optimización para un objetivo empresarial: en el caso de algunas recetas de grupos de conjuntos de datos personalizados, puede optimizar una solución para un objetivo personalizado, como maximizar los minutos de streaming o aumentar los ingresos. Para obtener más información, consulte Optimización de una solución para un objetivo adicional.
- Filtrado de recomendaciones: utilice filtros para aplicar reglas empresariales a las recomendaciones. Puede utilizar filtros para incluir o excluir determinados tipos de elementos de las recomendaciones. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Creación de un grupo de conjuntos de datos de Amazon Personalize

Después de crear archivos JSON de esquema para los datos, podrá crear un grupo de conjuntos de datos. En Amazon Personalize, un grupo de conjuntos de datos es un contenedor para los recursos de Amazon Personalize, incluidos los conjuntos de datos, los recomendadores de dominios y los recursos personalizados. Un grupo de conjuntos de datos organiza sus recursos en colecciones independientes, de modo que los recursos de un grupo de conjuntos de datos no pueden influir en los de ningún otro grupo de conjuntos de datos.

Puede crear un grupo de conjunto de datos para cada uno de sus dominios de negocio. Por ejemplo, es posible que tenga una aplicación que proporciona recomendaciones para el streaming en vídeo y otra que ofrece recomendaciones para audiolibros. En Amazon Personalize, crearía un grupo de conjuntos de datos para cada aplicación. De esta forma, los datos de una aplicación no influyen en las recomendaciones que Amazon Personalize genera para la otra aplicación.

Puede crear un grupo de conjuntos de datos de dominio o un grupo de conjuntos de datos personalizado:

- Con un Grupo de conjuntos de datos de dominio, se crean recursos preconfigurados y optimizados para distintos casos de uso. Cuando crea un grupo de conjuntos de datos, lo convierte en un grupo de conjuntos de datos de dominio especificando un dominio de VIDEO_ON_DEMAND o ECOMMERCE.

  Si tiene una aplicación de streaming en vídeo o de comercio electrónico, recomendamos que cree un grupo de conjuntos de datos de dominio. Aún puede añadir recursos personalizados, como soluciones y versiones de soluciones entrenadas para casos de uso personalizados. No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio.
- Un Grupo de conjuntos de datos personalizados solo incluye los recursos personalizados que se configuran en función del caso de uso. Con los recursos personalizados, puede entrenar e implementar soluciones configurables y versiones de soluciones (un modelo de recomendación de Amazon Personalize entrenado) en función de las necesidades de su empresa. Si no tiene una aplicación de VIDEO_ON_DEMAND o ECOMMERCE, le recomendamos que cree un grupo de conjuntos de datos personalizado. De lo contrario, recomendamos empezar con un grupo de conjuntos de datos de dominio y añadir recursos personalizados según sea necesario.

Puede crear un grupo de conjuntos de datos con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Temas

- Creación de un grupo de conjuntos de datos (consola)
- Creación de un grupo de conjuntos de datos (AWS CLI)
- Creación de un grupo de conjuntos de datos (SDK de AWS)

Creación de un grupo de conjuntos de datos (consola)

Cree un grupo de conjuntos de datos especificando el nombre del grupo de conjuntos de datos en la consola de Amazon Personalize.

Para crear un grupo de conjuntos de datos.

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Elija Crear grupo de conjuntos de datos.
3. Si es la primera vez que utiliza Amazon Personalize, en la página Crear grupo de conjunto de datos, en Nuevo grupo de conjunto de datos, elija Comenzar.
4. En Detalles del grupo de conjuntos de datos, en Nombre del grupo de conjuntos de datos, especifique un nombre para el grupo de conjuntos de datos.
5. Seleccione su Dominio:
   - Elija Comercio electrónico para crear un grupo de conjuntos de datos de dominio de ECOMMERCE.
   - Elija Vídeo en diferido para crear un grupo de conjuntos de datos de dominio de VIDEO_ON_DEMAND.
   - Elija Personalizado para crear un grupo de conjuntos de datos personalizado con solo recursos personalizados, como soluciones, campañas y trabajos de inferencia por lotes.
6. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
7. Elija Crear grupo de conjuntos de datos. Se abre la página Información general. Ya puede crear un esquema y un conjunto de datos. Consulte Creación de un esquema y un conjunto de datos.

Creación de un grupo de conjuntos de datos (AWS CLI)

Para crear un grupo de conjuntos de datos, utilice la operación `create-dataset-group`. Para crear un grupo de conjuntos de datos de dominio, para el dominio especifique `ECOMMERCE` o `VIDEO_ON_DEMAND`. Para crear un grupo de conjuntos de datos personalizado, no especifique un dominio. Puede utilizar el parámetro Etiquetas para etiquetar, de forma opcional, los recursos en Amazon Personalize. Para ver un ejemplo, consulte Adición de etiquetas (AWS CLI).

El siguiente código crea un grupo de conjuntos de datos de dominio para el dominio `VIDEO_ON_DEMAND`.

```
aws personalize create-dataset-group \
--name dataset-group-name \
--domain VIDEO_ON_DEMAND
```

Si se ejecuta correctamente, el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos se muestra de la siguiente manera.

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

Registre este valor para su uso futuro. Para visualizar el grupo de conjunto de datos que ha creado, utilice el comando `describe-dataset-group` y especifique el ARN del grupo de conjuntos de datos devuelto.

```
aws personalize describe-dataset-group \
--dataset-group-arn dataset group arn
```

El grupo de conjuntos de datos y sus propiedades se muestran de la siguiente manera.

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

```
        "lastUpdatedDateTime": 1542396513.377
    }
}
```

Cuando el estado del grupo de conjuntos de datos `status` sea ACTIVE, podrá crear un esquema y un conjunto de datos. Consulte Creación de un esquema y un conjunto de datos.

Creación de un grupo de conjuntos de datos (SDK de AWS)

Cree un grupo de conjuntos de datos de dominio con el siguiente código. Asigne un nombre al grupo de conjuntos de datos de dominio y, para `domain`, especifique `ECOMMERCE` o `VIDEO_ON_DEMAND`. Para crear un grupo de conjuntos de datos personalizado, modifique el código para eliminar el parámetro de dominio.

Para obtener más información sobre estas operaciones de la API, consulte CreateDatasetGroup en la sección de referencia de la API. Puede utilizar el parámetro Etiquetas para etiquetar, de forma opcional, los recursos en Amazon Personalize. Para ver un ejemplo, consulte Adición de etiquetas (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();
```

La operación DescribeDatasetGroup devuelve el `datasetGroupArn` y el estado de la operación. Cuando el estado del grupo de conjuntos de datos `status` sea ACTIVE, podrá crear un esquema y un conjunto de datos. Consulte Creación de un esquema y un conjunto de datos.

Creación de un esquema y un conjunto de datos

Después de crear un grupo de conjuntos de datos, podrá crear un esquema de Amazon Personalize y un conjunto de datos para cada tipo de datos que vaya a importar. Un esquema informa a Amazon Personalize de la estructura de sus datos y permite que Amazon Personalize analice los datos. Al crear un esquema en Amazon Personalize, debe usar el archivo JSON que ha creado en Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Un conjunto de datos es un contenedor de datos de entrenamiento en Amazon Personalize. Los distintos tipos de conjuntos de datos tienen requisitos diferentes. Debe crear un conjunto de datos para cada tipo de datos que vaya a importar. Para obtener información sobre los distintos tipos de conjuntos de datos y cómo preparar los datos, consulte Preparación de los datos de entrenamiento para Amazon Personalize.

Puede crear esquemas y conjuntos de datos con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o. AWS SDKs No puede crear recursos de siguiente mejor acción, incluidos los conjuntos de datos de acciones e interacciones de acciones, en un grupo de conjuntos de datos de dominio.

> **Important**
>
> Una vez que se haya creado un esquema, no se podrán realizar cambios en él. Sin embargo, si agrega columnas nuevas, puede reemplazar el esquema de un conjunto de datos por uno nuevo. Para obtener más información, consulte Reemplazo del esquema de un conjunto de datos para agregar nuevas columnas.

Temas

- Creación de un conjunto de datos y un esquema (consola)
- Creación de un conjunto de datos y un esquema (AWS CLI)
- Creación de un conjunto de datos y un esquema (AWS SDKs)

Creación de un conjunto de datos y un esquema (consola)

Si este es el primer conjunto de datos de su grupo de conjuntos de datos, su primer tipo de conjunto de datos será un conjunto de datos de interacciones de elementos. Para crear el conjunto de datos

de interacciones de elementos en la consola, especifique el nombre del conjunto de datos y, a continuación, especifique un esquema JSON en formato Avro. Si no es el primer conjunto de datos de este grupo de conjuntos de datos, elija el tipo de conjunto de datos y, a continuación, especifique un nombre y un esquema.

Para obtener información sobre los requisitos de conjuntos de datos de Amazon Personalize, consulte Preparación de los datos de entrenamiento para Amazon Personalize. Si acaba de completar Creación de un grupo de conjuntos de datos de Amazon Personalize y ya está creando el conjunto de datos, vaya al paso 4 de este procedimiento.

Para crear un conjunto de datos y un esquema

1. Si aún no lo ha hecho, siga las instrucciones de Creación de archivos JSON de esquema para los esquemas de Amazon Personalize para crear un archivo JSON de esquema que describa los datos.
2. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
3. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos que creó en Creación de un grupo de conjuntos de datos de Amazon Personalize.
4. En Paso 1. Crear conjuntos de datos e importar datos, elija Crear conjunto de datos y seleccione el tipo de conjunto de datos que desea crear.
5. Elija Importar datos directamente a los conjuntos de datos de Amazon Personalize y elija Siguiente.
6. En Detalles del conjunto de datos, en Nombre del conjunto de datos, especifique un nombre para el conjunto de datos.
7. En Esquema de conjunto de datos, elija Crear un esquema nuevo o Usar un esquema existente.
8. Si está utilizando un esquema existente, elija el esquema existente que se usará. Si va a crear un esquema nuevo, asigne un nombre al esquema y pegue el código JSON del esquema que coincida con los datos. Ha creado este archivo en Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.
9. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
10. Elija Siguiente y siga las instrucciones de Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize para importar los datos.

Creación de un conjunto de datos y un esquema (AWS CLI)

Para crear un conjunto de datos y un esquema utilizando el AWS CLI`create-schema` comando the (que usa la operación CreateSchema API) y luego `create-dataset` (que usa la operación CreateDataset API).

Para crear un esquema y un conjunto de datos

1. Si aún no lo ha hecho, siga las instrucciones de Creación de archivos JSON de esquema para los esquemas de Amazon Personalize para crear un archivo JSON de esquema que describa los datos.
2. Ejecute el siguiente comando para crear un esquema en Amazon Personalize. Una vez que se haya creado un esquema, no se podrán realizar cambios en él. Reemplace `schemaName` por el nombre del esquema y `file://SchemaName.json` por la ubicación del archivo JSON. En el ejemplo, el archivo pertenece a la carpeta actual. Si va a crear un esquema para un conjunto de datos en un grupo de conjunto de datos de dominio, debe agregar el parámetro `domain` y establecerlo en `ECOMMERCE` o `VIDEO_ON_DEMAND`. Para obtener más información sobre la API, consulte CreateSchema.

   ```
   aws personalize create-schema \
     --name SchemaName \
     --schema file://SchemaName.json
   ```

   Se muestra el nombre de recurso de Amazon (ARN) del esquema, como se muestra en el siguiente ejemplo:

   ```
   {
     "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/SchemaName"
   }
   ```
3. Ejecute el siguiente comando para crear un conjunto de datos vacío. Proporcione el grupo de conjunto de datos del que proviene el nombre de recurso de Amazon (ARN) Creación de un grupo de conjuntos de datos (AWS CLI) y el esquema de ARN del paso anterior. Los valores del tipo de conjunto de datos pueden ser `Interactions`, `Users`, `Items`, `Actions` o `Action_Interactions`. Para obtener más información sobre la API, consulte CreateDataset.

   ```
   aws personalize create-dataset \
     --name Dataset Name \
     --dataset-group-arn Dataset Group ARN \
   ```

```
  --dataset-type Dataset Type \
  --schema-arn Schema Arn
```

Se muestra el ARN del conjunto de datos, tal como se muestra en el ejemplo siguiente.

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

4. Registre el ARN del conjunto de datos para su uso posterior. Una vez que haya creado un conjunto de datos, estará listo para importar sus datos de entrenamiento. Consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.

Creación de un conjunto de datos y un esquema (AWS SDKs)

Para crear un conjunto de datos y un esquema mediante el AWS SDKs, primero debe definir un esquema en formato Avro y añadirlo a Amazon Personalize mediante la CreateSchema operación. Una vez que se haya creado un esquema, no se podrán realizar cambios en él. A continuación, cree un conjunto de datos mediante la operación CreateDataset.

Para crear un esquema y un conjunto de datos

1. Si aún no lo ha hecho, siga las instrucciones de Creación de archivos JSON de esquema para los esquemas de Amazon Personalize para crear un archivo JSON de esquema que describa los datos.
2. Cree un esquema en Amazon Personalize con el siguiente código. Especifique el nombre de su esquema y la ruta del archivo JSON del esquema. Si va a crear un esquema para un conjunto de datos en un grupo de conjunto de datos de dominio, debe agregar el parámetro `domain` y establecerlo en `ECOMMERCE` o `VIDEO_ON_DEMAND`. Para obtener más información sobre la API, consulte 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 devuelve el ARN del nuevo esquema. Anote el valor, ya que lo necesitará en el siguiente paso.

3. Cree un conjunto de datos utilizando la operación CreateDataset. El código siguiente muestra cómo crear un conjunto de datos. Especifique el nombre de recurso de Amazon (ARN) de su grupo de conjuntos de datos, el ARN del esquema del paso anterior y especifique el tipo de conjunto de datos. Los valores del tipo de conjunto de datos pueden ser `Interactions`, `Users`, `Items`, `Actions` o `Action_Interactions`. Para obtener información sobre los distintos tipos de conjuntos de datos, consulte Preparación de los datos de entrenamiento para 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();
```

Una vez que haya creado un conjunto de datos, estará listo para importar sus datos de entrenamiento. Consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.

Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize

Después de completar la creación de un esquema y un conjunto de datos, podrá importar los datos de entrenamiento al conjunto de datos. Al importar datos, puede optar por importar los registros de forma masiva, individual o de ambos modos.

- Las importaciones masivas implican la importación de una gran cantidad de registros históricos a la vez. Puede preparar datos masivos e importarlos directamente en Amazon Personalize desde un archivo CSV en Amazon S3. Para obtener información sobre cómo preparar los datos, consulte Preparación de los datos de entrenamiento para Amazon Personalize. Si necesitas ayuda para preparar tus datos, puedes usar SageMaker AI Data Wrangler para preparar e importar tus datos masivos de interacción de artículos, usuarios y artículos. Para obtener más información, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.
- Si no tiene datos masivos, puede utilizar operaciones de importación individuales para recopilar datos y transmitir eventos hasta que cumpla los requisitos de entrenamiento de Amazon Personalize y los requisitos de datos del caso de uso o receta del dominio. Para obtener información sobre el registro de eventos, consulte Registro de los eventos en tiempo real para influir en las recomendaciones. Para obtener información sobre la importación de registros individuales, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize.

Tras importar los datos a un conjunto de datos de Amazon Personalize, puede analizarlos, exportarlos a un bucket de Amazon S3, actualizarlos o eliminarlos mediante la eliminación del conjunto de datos.

Si importa un artículo, usuario o acción con el mismo ID que otro registro que ya se encuentra en su conjunto de datos, Amazon Personalize lo sustituye por el nuevo registro. Si registra dos eventos de interacción con artículos o de interacción de acción con la misma marca de tiempo e idénticas propiedades, Amazon Personalize solo conserva uno de ellos.

A medida que su catálogo crezca, actualice sus datos históricos con operaciones adicionales de importación de datos masivos o individuales. Para obtener recomendaciones en tiempo real, mantenga su conjunto de datos de interacciones de elementos actualizado según el comportamiento de los usuarios. Para ello, registre los eventos de interacción en tiempo real con un rastreador de

eventos y la operación PutEvents. Para obtener más información, consulte Registro de los eventos en tiempo real para influir en las recomendaciones

Después de importar los datos, podrá crear recomendadores de dominios (para grupos de conjuntos de datos de dominio) o recursos personalizados (para grupos de conjuntos de datos personalizados) para entrenar un modelo a partir de los datos. Usará estos recursos para generar recomendaciones. Para obtener más información, consulte Recomendadores de dominios en Amazon Personalize o Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Temas

- Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos
- Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler
- Importación de registros individuales a un conjunto de datos de Amazon Personalize

Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos

Después de formatear los datos de entrada (consulte Preparación de los datos de entrenamiento para Amazon Personalize) y completar Creación de un esquema y un conjunto de datos, podrá importar los datos masivos con un trabajo de importación de conjuntos de datos. Un trabajo de importación de conjuntos de datos es una herramienta de importación masiva que rellena un conjunto de datos con datos de Amazon S3.

Para importar datos de Amazon S3, los archivos CSV deben estar en un bucket de Amazon S3 y debe conceder permiso a Amazon Personalize para acceder a los recursos de Amazon S3:

- Para obtener información sobre la carga de archivos en Amazon S3, consulte Carga de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.
- Para obtener información sobre la concesión de acceso a Amazon Personalize a los archivos en Amazon S3, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

  Si utilizas AWS Key Management Service (AWS KMS) para el cifrado, debes conceder permiso a Amazon Personalize y a tu rol de servicio Amazon Personalize IAM para usar tu clave. Para

obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Puede crear un trabajo de importación de conjuntos de datos mediante la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS SDKs. Si anteriormente ha creado un trabajo de importación de conjuntos de datos para un conjunto de datos, puede usar un nuevo trabajo de ese tipo para agregar o reemplazar los datos masivos existentes. Para obtener más información, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Si importa un artículo, usuario o acción con el mismo ID que otro registro que ya se encuentra en su conjunto de datos, Amazon Personalize lo sustituye por el nuevo registro. Si registra dos eventos de interacción con artículos o de interacción de acción con la misma marca de tiempo e idénticas propiedades, Amazon Personalize solo conserva uno de ellos.

Después de importar los datos, podrá crear recomendadores de dominios (para grupos de conjuntos de datos de dominio) o recursos personalizados (para grupos de conjuntos de datos personalizados) para entrenar un modelo a partir de los datos. Usará estos recursos para generar recomendaciones. Para obtener más información, consulte Recomendadores de dominios en Amazon Personalize o Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Temas

- Modos de importación
- Creación de un trabajo de exportación de conjuntos de datos (consola)
- Creación de un trabajo de importación de conjuntos de datos (AWS CLI)
- Creación de un trabajo de importación de conjuntos de datos (AWS SDKs)

Modos de importación

Si ya ha creado un trabajo de importación para el conjunto de datos, puede configurar la forma en que Amazon Personalize agrega los nuevos registros. Para ello, especifique un modo de importación para el trabajo de importación de conjuntos de datos. Si no ha importado registros masivos, el campo Modo de importación no está disponible en la consola y solo puede especificar `FULL` en la operación de la API `CreateDatasetImportJob`. El valor predeterminado es un reemplazo completo.

- Para sobrescribir todos los datos masivos existentes en su conjunto de datos, elija Reemplazar los datos existentes en la consola de Amazon Personalize o especifique FULL en la operación de la API CreateDatasetImportJob. Esto no reemplaza los datos que haya importado individualmente, incluidos los eventos registrados en tiempo real.
- Para anexar los registros a los datos existentes en su conjunto de datos, elija Agregar a los datos existentes o especifique `INCREMENTAL` en la operación de la API `CreateDatasetImportJob`. Amazon Personalize reemplaza cualquier registro con el mismo identificador por el nuevo.

> **Note**
>
> Para anexar datos a un conjunto de datos de interacciones de elementos o de interacciones de acción con un trabajo de importación de conjuntos de datos, debe tener al menos 1000 registros de interacciones de elementos o de interacciones de acción nuevos.

Creación de un trabajo de exportación de conjuntos de datos (consola)

> **Important**
>
> De forma predeterminada, un trabajo de importación de conjuntos de datos reemplaza cualquier dato existente en el conjunto de datos que haya importado de manera masiva. Si ya ha importado datos masivos, puede adjuntar datos si cambia el modo de importación del trabajo.

Para importar registros masivos en un conjunto de datos con la consola de Amazon Personalize, cree un trabajo de importación de conjuntos de datos con un nombre, el rol de servicio de IAM y la ubicación de sus datos.

Si acaba de crear su conjunto de datos en Creación de un esquema y un conjunto de datos, vaya al paso 5.

Para importar registros masivos (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos. Aparece la Información general del grupo de conjuntos de datos.

3. En el panel de navegación, elija Conjuntos de datos y seleccione el conjunto de datos en el que quiere importar datos masivos.
4. En Trabajos de importación de conjuntos de datos, seleccione Crear trabajo de importación de conjuntos de datos.
5. Si es el primer trabajo de importación de conjuntos de datos, en Origen de importación de datos, elija Importar datos de S3.
6. En Nombre del trabajo de importación del conjunto de datos, especifique un nombre para su trabajo de importación.
7. Si ya ha importado datos masivos, en Modo de importación, elija cómo actualizar el conjunto de datos. Elija Reemplazar datos existentes o Agregar a datos existentes. Esta opción no aparece si es el primer trabajo para el conjunto de datos. Para obtener más información, consulte Actualización de datos en conjuntos de datos después del entrenamiento.
8. En Origen de importación de datos, en Ubicación de datos, especifique dónde está almacenado el archivo de datos en Amazon S3. Utilice la siguiente sintaxis:

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

   Si sus archivos CSV se encuentran en una carpeta de su bucket de Amazon S3 y desea subir varios archivos CSV a un conjunto de datos con un solo trabajo de importación de conjuntos de datos, puede especificar la ruta a la carpeta. Amazon Personalize solo utiliza los archivos del primer nivel de su carpeta, no utiliza los datos de ninguna subcarpeta. Use la siguiente sintaxis con una / después del nombre de la carpeta:

   ```
   s3:/amzn-s3-demo-bucket/<folder path>/
   ```
9. En Rol de IAM, elija crear un rol nuevo o usar uno existente. Si ha completado los requisitos, elija Usar un rol de servicio existente y especifique el rol que creó en Creación de un rol de IAM para Amazon Personalize.
10. Si ha creado una atribución de métricas y desea publicar las métricas relacionadas con este trabajo en Amazon S3, en Publicar métricas de eventos en S3, elija Publicar métricas para este trabajo de importación.

    Si no ha creado una y desea publicar métricas para este trabajo, elija Crear atribución de métricas para crear una nueva en una pestaña diferente. Después de crear la atribución métrica, puede volver a esta pantalla y terminar de crear el trabajo de importación.

    Para obtener más información sobre atribuciones de métricas, consulte Medición del impacto de las recomendaciones de Amazon Personalize.

11. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
12. Elija Start import (Iniciar importación). Se iniciará el trabajo de importación de los datos y aparecerá la página Información general del panel. La importación del conjunto de datos finaliza cuando el estado es ACTIVO. Tras importar los datos a un conjunto de datos de Amazon Personalize, puede analizarlos, exportarlos a un bucket de Amazon S3, actualizarlos o eliminarlos suprimiendo el conjunto de datos.

    Después de importar los datos, podrá crear recomendadores de dominios (para grupos de conjuntos de datos de dominio) o recursos personalizados (para grupos de conjuntos de datos personalizados) para entrenar un modelo a partir de los datos. Usará estos recursos para generar recomendaciones. Para obtener más información, consulte Recomendadores de dominios en Amazon Personalize o Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Creación de un trabajo de importación de conjuntos de datos (AWS CLI)

> **Important**
>
> De forma predeterminada, un trabajo de importación de conjuntos de datos reemplaza cualquier dato existente en el conjunto de datos que haya importado de manera masiva. Si ya ha importado datos masivos, puede adjuntar datos si cambia el modo de importación del trabajo.

Para importar registros masivos mediante el AWS CLI, cree un trabajo de importación de conjuntos de datos mediante el CreateDatasetImportJob comando. Si anteriormente creó un trabajo de importación de conjuntos de datos para un conjunto de datos, puede usar el parámetro de modo de importación para especificar cómo agregar los nuevos datos. Para obtener más información sobre la actualización de los datos masivos existentes, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Importar registros masivos (AWS CLI)

1. Ejecute el siguiente comando para crear un trabajo de importación de conjunto de datos. Proporcione el Nombre de recurso de Amazon (ARN) del conjunto de datos y especifique la ruta

al bucket de Amazon S3 donde almacenó los datos de entrenamiento. Use la siguiente sintaxis para la ruta:

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

Si sus archivos CSV se encuentran en un bucket de Amazon S3 y desea subir varios archivos CSV a un conjunto de datos con un solo trabajo de importación de conjuntos de datos, puede especificar la ruta a la carpeta. Amazon Personalize solo utiliza los archivos del primer nivel de su carpeta, no utiliza los datos de ninguna subcarpeta. Use la siguiente sintaxis con una `/` después del nombre de la carpeta:

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

Indique el rol AWS Identity and Access Management (IAM) Amazon Resource Name (ARN) en el que creó. Creación de un rol de IAM para Amazon Personalize El `import-mode` predeterminado es `FULL`. Para obtener más información, consulte Actualización de datos en conjuntos de datos después del entrenamiento. Para obtener más información acerca de la operación, consulte 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
```

Se muestra el ARN del trabajo de importación del conjunto de datos, como se muestra en el siguiente ejemplo.

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

2. Compruebe el estado utilizando el comando `describe-dataset-import-job`. Proporcione el ARN del trabajo de importación de conjunto de datos que obtuvo en el paso anterior. Para obtener más información acerca de la operación, consulte DescribeDatasetImportJob.

```
aws personalize describe-dataset-import-job \
```

```
--dataset-import-job-arn dataset import job arn
```

Se muestran las propiedades del trabajo de importación del conjunto de datos y su estado. En un principio, el valor de `status` es CREACIÓN PENDIENTE.

```
{
  "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
  }
}
```

La importación del conjunto de datos finaliza cuando el estado es ACTIVO. Tras importar los datos a un conjunto de datos de Amazon Personalize, puede analizarlos, exportarlos a un bucket de Amazon S3, actualizarlos o eliminarlos suprimiendo el conjunto de datos.

Después de importar los datos, podrá crear recomendadores de dominios (para grupos de conjuntos de datos de dominio) o recursos personalizados (para grupos de conjuntos de datos personalizados) para entrenar un modelo a partir de los datos. Usará estos recursos para generar recomendaciones. Para obtener más información, consulte Recomendadores de dominios en Amazon Personalize o Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Creación de un trabajo de importación de conjuntos de datos (AWS SDKs)

> **Important**
>
> De forma predeterminada, un trabajo de importación de conjuntos de datos reemplaza cualquier dato existente en el conjunto de datos que haya importado de manera masiva. Si ya ha importado datos masivos, puede adjuntar datos si cambia el modo de importación del trabajo.

Para importar datos, cree un trabajo de importación de conjuntos de datos con la operación CreateDatasetImportJob. En el siguiente código se muestra cómo crear un trabajo de importación de conjuntos de datos.

Indique el nombre del trabajo, establezca el Nombre de recurso de Amazon (ARN) `datasetArn` de su conjunto de datos y establezca la `dataLocation` a la ruta para su bucket de Amazon S3 donde almacenó los datos de entrenamiento. Use la siguiente sintaxis para la ruta:

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

Si sus archivos CSV se encuentran en un bucket de Amazon S3 y desea subir varios archivos CSV a un conjunto de datos con un solo trabajo de importación de conjuntos de datos, puede especificar la ruta a la carpeta. Amazon Personalize solo utiliza los archivos del primer nivel de su carpeta, no utiliza los datos de ninguna subcarpeta. Use la siguiente sintaxis con una / después del nombre de la carpeta:

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

En el caso de`roleArn`, especifique el rol AWS Identity and Access Management (IAM) que otorga a Amazon Personalize los permisos para acceder a su bucket de S3. Consulte Creación de un rol de IAM para Amazon Personalize. El `importMode` predeterminado es `FULL`. Esto reemplaza todos los datos masivos del conjunto de datos. Para adjuntar datos, establézcalos a `INCREMENTAL`. Para obtener más información sobre la actualización de los datos masivos existentes, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

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 respuesta de la operación DescribeDatasetImportJob incluye el estado de la operación.

Debe esperar hasta que el estado cambie a ACTIVO para poder utilizar los datos para entrenar un modelo.

La importación del conjunto de datos finaliza cuando el estado es ACTIVO. Tras importar los datos a un conjunto de datos de Amazon Personalize, puede analizarlos, exportarlos a un bucket de Amazon S3, actualizarlos o eliminarlos suprimiendo el conjunto de datos.

Después de importar los datos, podrá crear recomendadores de dominios (para grupos de conjuntos de datos de dominio) o recursos personalizados (para grupos de conjuntos de datos personalizados) para entrenar un modelo a partir de los datos. Usará estos recursos para generar recomendaciones. Para obtener más información, consulte Recomendadores de dominios en Amazon Personalize o Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler

> **Important**
>
> AI usar Data Wrangler, incurre en costos de IA. SageMaker Para obtener una lista completa de cargos y precios, consulta la pestaña Data Wrangler de los precios de Amazon SageMaker AI. Para evitar incurrir en cargos adicionales, cuando haya terminado, cierre su instancia de Data Wrangler. Para obtener más información, consulte Apagar Data Wrangler.

Después de crear un grupo de conjuntos de datos, puede usar Amazon SageMaker AI Data Wrangler (Data Wrangler) para importar datos de más de 40 fuentes a un conjunto de datos de Amazon Personalize. Data Wrangler es una función de Amazon SageMaker AI Studio Classic que proporciona una solución integral para importar, preparar, transformar y analizar datos. No se puede usar Data Wrangler para preparar e importar datos en un conjunto de datos de acciones ni en un conjunto de datos de interacciones de acción.

Cuando utiliza Data Wrangler para preparar e importar datos, emplea un flujo de datos. Un flujo de datos define una serie de pasos de preparación de datos mediante machine learning, empezando por la importación de datos. Cada vez que añade un paso a su flujo, Data Wrangler realiza una acción con sus datos, como transformarlos o generar una visualización.

Estos son algunos de los pasos que puede añadir a su flujo para preparar datos para Amazon Personalize:

- Estadísticas: puede añadir pasos de información específicos de Amazon Personalize a su flujo. Esta información puede ayudarle a conocer sus datos y a saber qué medidas puede llevar a cabo para mejorarlos.
- Visualizaciones: puede añadir pasos de visualización para generar gráficos, como histogramas y gráficos de dispersión. Los gráficos pueden ayudarle a detectar problemas en los datos, como valores atípicos o valores que faltan.
- Transformaciones: puede utilizar los pasos de transformación específicos y generales de Amazon Personalize para asegurarse de que sus datos cumplen los requisitos de Amazon Personalize. La transformación de Amazon Personalize le ayuda a asignar sus columnas de datos a las columnas obligatorias en función del tipo de conjunto de datos de Amazon Personalize.

Si necesita salir de Data Wrangler antes de importar datos en Amazon Personalize, puede volver a donde los dejó eligiendo el mismo tipo de conjunto de datos al iniciar Data Wrangler desde la consola de Amazon Personalize. O bien, puede acceder a Data Wrangler directamente a través SageMaker de AI Studio Classic.

Recomendamos que importe los datos de Data Wrangler en Amazon Personalize de la siguiente manera. Los pasos de transformación, visualización y análisis son opcionales, repetibles y se pueden completar en cualquier orden.

1. Configurar permisos: configure los permisos para las funciones de servicio Amazon Personalize y SageMaker AI. Y configure los permisos para sus usuarios.
2. Inicie Data Wrangler en SageMaker AI Studio Classic desde la consola Amazon Personalize: utilice la consola Amazon Personalize para configurar un dominio de SageMaker IA e inicie Data Wrangler en SageMaker AI Studio Classic.
3. Importar sus datos en Data Wrangler: importe datos de más de 40 orígenes en Data Wrangler. Las fuentes incluyen AWS servicios, como Amazon Redshift, Amazon EMR o Amazon Athena, y terceros, como Snowflake o. DataBricks
4. Transformar sus datos: utilice Data Wrangler para transformar sus datos y cumplir con los requisitos de Amazon Personalize.
5. Visualizar y analizar sus datos: utilice Data Wrangler para visualizar sus datos y analizarlos mediante información específica de Amazon Personalize.
6. Procese e importe datos a Amazon Personalize: utilice una libreta Jupyter clásica de SageMaker AI Studio para importar los datos procesados a Amazon Personalize.

Información adicional

Los siguientes recursos proporcionan información adicional sobre el uso de Amazon SageMaker AI Data Wrangler y Amazon Personalize.

- Para ver un tutorial que le guiará a través del procesamiento y la transformación de un conjunto de datos de muestra, consulte Demo: Tutorial del conjunto de datos Data Wrangler Titanic en la Guía para desarrolladores de Amazon SageMaker AI. Este tutorial presenta los campos y las funciones de Data Wrangler.
- Para obtener información sobre la incorporación a los dominios de Amazon SageMaker AI, consulte Incorporación rápida a Amazon SageMaker AI Domain en la Guía para desarrolladores de Amazon SageMaker AI.
- Para obtener información sobre los requisitos de datos de Amazon Personalize, consulte Preparación de los datos de entrenamiento para Amazon Personalize.

Configuración de permisos

Para preparar los datos con Data Wrangler, debe configurar los siguientes permisos:

- Crear un rol de servicio para Amazon Personalize: si aún no lo ha hecho, complete las instrucciones de Configuración de Amazon Personalize para crear un rol de servicio de IAM para Amazon Personalize. Este rol debe tener los permisos `GetObject` y `ListBucket` para los buckets de Amazon S3 que almacenan sus datos procesados. Además, debe tener permiso para usar cualquier AWS KMS clave.

  Para obtener información sobre la concesión de acceso a Amazon Personalize a sus buckets de Amazon S3, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3. Para obtener información sobre cómo conceder a Amazon Personalize acceso a tus AWS KMS claves, consultaOtorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
- Cree un usuario administrativo con permisos de SageMaker IA: su administrador debe tener acceso completo a la SageMaker IA y debe poder crear un dominio de SageMaker IA. Para obtener más información, consulte Crear un usuario y un grupo administrativos en la Guía para desarrolladores de Amazon SageMaker AI.
- Cree una función de ejecución de SageMaker IA: cree una función de ejecución de SageMaker IA con acceso a los recursos de SageMaker IA y a las operaciones de importación de datos de Amazon Personalize. La función de ejecución de la SageMaker IA debe tener la `AmazonSageMakerFullAccess`política adjunta. Si necesita permisos de Data Wrangler más

detallados, consulte Seguridad y permisos de Data Wrangler en la Guía para desarrolladores de Amazon SageMaker AI. Para obtener más información sobre las funciones de SageMaker IA, consulte Funciones de IA. SageMaker

Para conceder acceso a las operaciones de importación de datos de Amazon Personalize, adjunte la siguiente política de IAM a la función de ejecución de SageMaker IA. Esta política otorga los permisos necesarios para importar datos a Amazon Personalize y asociar una política a su bucket de Amazon S3. Y concede permisos de `PassRole` cuando el servicio es Amazon Personalize. Actualice el `amzn-s3-demo-bucket` de Amazon S3 con el nombre del bucket de Amazon S3 que desee usar como destino para sus datos con formato después de prepararlos con 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"
                }
            }
        }
    ]
}
```

Para obtener más información sobre la creación de una política de IAM, consulte Crear políticas de IAM en la Guía del usuario de IAM. Para obtener información sobre cómo asociar una política de IAM a rol, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

Lanzamiento de Data Wrangler desde Amazon Personalize

Para lanzar Data Wrangler desde Amazon Personalize, utilice la consola Amazon Personalize para configurar un dominio de SageMaker IA e iniciar Data Wrangler.

Para lanzar Data Wrangler desde Amazon Personalize

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En Configurar conjuntos de datos, elija Crear conjunto de datos y seleccione el tipo de conjunto de datos que quiera crear. No se puede usar Data Wrangler para preparar un conjunto de datos de acciones ni un conjunto de datos de interacciones de acción.
4. Elija Importar datos con Data Wrangler y, después, Siguiente.
5. Para SageMaker el dominio, elige usar un dominio existente o crear uno nuevo. Necesitas un dominio de SageMaker IA para acceder a Data Wrangler en SageMaker AI Studio Classic. Para obtener información sobre los dominios y los perfiles de usuario, consulte SageMaker AI Domain en la Guía para desarrolladores de Amazon SageMaker AI.
6. Para usar un dominio existente, elija un dominio de SageMaker IA y un perfil de usuario para configurar el dominio.
7. Para crear un nuevo dominio:
   - Asigne un nombre al nuevo dominio.
   - Elija un Nombre de perfil de usuario.

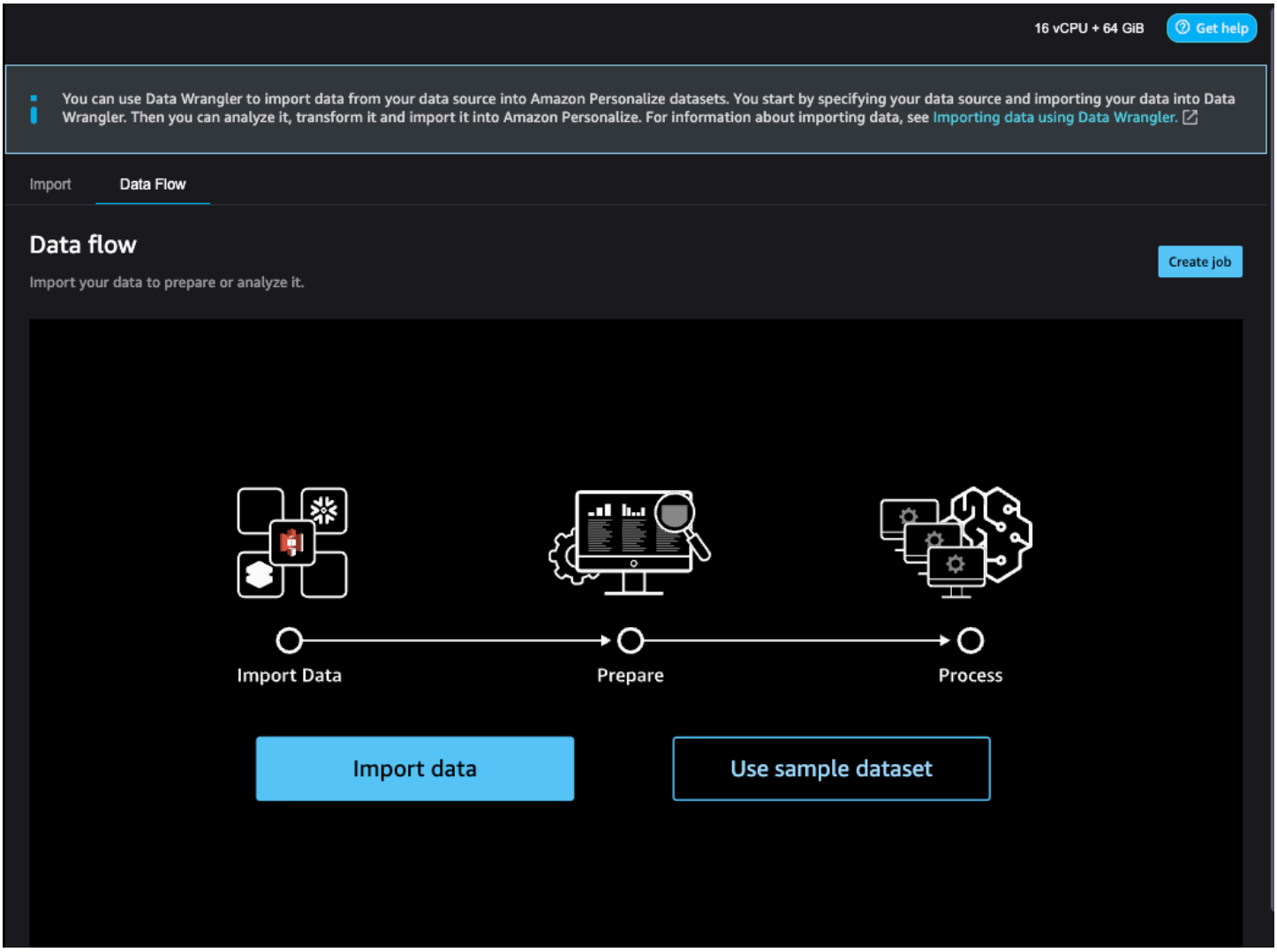
- En Rol de ejecución, elija el rol que creó en Configuración de permisos. O bien, si tiene CreateRole permisos, cree un nuevo rol mediante el asistente de creación de roles. El rol que use debe tener la política `AmazonSageMakerFullAccess` asociada.

8. Elija Siguiente. Si vas a crear un dominio nuevo, la SageMaker IA empezará a crear el tuyo. Este proceso puede tardar hasta 10 minutos.
9. Revisa los detalles de tu dominio de SageMaker IA.
10. Elige Importar datos con Data Wrangler. SageMaker AI Studio Classic comienza a crear tu entorno y, cuando haya terminado, la página de flujo de datos de Data Wrangler en SageMaker AI Studio Classic se abre en una nueva pestaña. SageMaker AI Studio Classic puede tardar hasta cinco minutos en terminar de crear tu entorno. Cuando termine, estará listo para empezar a importar datos en Data Wrangler. Para obtener más información, consulte Importación de datos en Data Wrangler.

Importación de datos en Data Wrangler

Tras configurar un dominio de SageMaker IA e iniciar Data Wrangler en una pestaña nueva, estará listo para importar los datos de su fuente a Data Wrangler. Al utilizar Data Wrangler para preparar datos para Amazon Personalize, importa un conjunto de datos cada vez. Recomendamos empezar por un conjunto de datos de interacciones de elementos. No se puede usar Data Wrangler para preparar un conjunto de datos de acciones ni un conjunto de datos de interacciones de acción.

Empiece en la página Flujo de datos. La página debería tener un aspecto similar al siguiente.



Para empezar a importar datos, elija Importar datos y especifique el origen de datos. Data Wrangler admite más de 40 orígenes. Estos incluyen AWS servicios, como Amazon Redshift, Amazon EMR o Amazon Athena, y de terceros, como Snowflake o. DataBricks Los distintos orígenes de datos tienen distintos procedimientos para conectar e importar datos.

Para obtener una lista completa de las fuentes disponibles e instrucciones paso a paso sobre la importación de datos, consulte Importación en la guía para desarrolladores de Amazon SageMaker AI.

Tras importar datos a Data Wrangler, estará listo para transformarlos. Para obtener información sobre la transformación de datos, consulte Transformación de datos.

Transformación de datos

Para transformar datos en Data Wrangler, añada un paso de transformación a su flujo de datos. Data Wrangler incluye más de 300 transformaciones que puede usar para preparar sus datos, incluida una transformación de Asignar columnas para Amazon Personalize. Además, puede usar las transformaciones generales de Data Wrangler para corregir problemas como valores atípicos, problemas de tipo y valores que faltan.

Cuando termine de transformar sus datos, podrá analizarlos con Data Wrangler. O bien, si ha terminado de preparar sus datos en Data Wrangler, puede procesarlos e importarlos en Amazon Personalize. Para obtener más información acerca del análisis de tablas, consulte Generación de visualizaciones y análisis de datos. Para obtener información sobre el procesamiento y la importación de datos, consulte Procesamiento de los datos y su importación en Amazon Personalize.

Temas

- Asignación de columnas para Amazon Personalize
- Transformaciones generales de Data Wrangler

Asignación de columnas para Amazon Personalize

Para transformar sus datos de manera que cumplan con los requisitos de Amazon Personalize, añada la transformación de Asignar columnas para Amazon Personalize y asigne las columnas a los campos obligatorios y opcionales para Amazon Personalize.

Para usar la transformación de Asignar columnas para Amazon Personalize

1. Elija + para su última transformación y Agregar transformación. Si no ha agregado ninguna transformación, elija el signo + para la transformación de Tipos de datos. Data Wrangler añade esta transformación automáticamente a su flujo.
2. Seleccione Agregar paso.
3. Elija Transformaciones para Amazon Personalize. La transformación de Asignar columnas para Amazon Personalize está seleccionada de manera predeterminada.
4. Utilice los campos de transformación para asignar sus datos a los atributos requeridos de Amazon Personalize.

    1. Elija el tipo de conjunto de datos que coincida con sus datos (interacciones, elementos o usuarios).

2. Elija su dominio (ECOMMERCE, VIDEO_ON_DEMAND o personalizado). El dominio que elija debe coincidir con el dominio que especificó al crear el grupo de conjuntos de datos.
3. Elija las columnas que coincidan con los campos obligatorios y opcionales de Amazon Personalize. Por ejemplo, para la columna Item_ID, elija la columna de sus datos que almacena la información de identificación única de cada uno de sus elementos.

   Cada campo de columna se filtra por tipo de datos. Solo están disponibles las columnas de sus datos que cumplan los requisitos de tipo de datos de Amazon Personalize. Si sus datos no son del tipo requerido, puede usar la transformación de Data Wrangler de Analizar valor como tipo para convertirlos.

Transformaciones generales de Data Wrangler

Las siguientes transformaciones generales de Data Wrangler pueden ayudarle a preparar los datos para Amazon Personalize:

- Conversión de tipos de datos: si su campo no aparece como una opción posible en la transformación de Asignar columnas para Amazon Personalize, es posible que necesite convertir su tipo de datos. La transformación de Data Wrangler Analizar valor como tipo puede ayudarle a convertir sus datos. O puede usar la transformación de Tipos de datos que Data Wrangler agrega de forma predeterminada cuando crea un flujo. Para usar esta transformación, elija el tipo de datos en las listas desplegables de Tipo, Vista previa y, a continuación, Actualizar.

  Para obtener información sobre los tipos de datos requeridos para los campos, consulte la sección correspondiente al tipo de dominio y conjunto de datos en Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.
- Gestión de valores faltantes y valores atípicos: si genera información sobre valores que faltan o valores atípicos, puede utilizar las transformaciones de Data Wrangler Gestionar valores atípicos y Gestionar valores que faltan para resolver estos problemas.
- Transformaciones personalizadas: con Data Wrangler, puede crear sus propias transformaciones con Python (User-Defined función) PySpark, pandas o PySpark (SQL). Puede usar una transformación personalizada para realizar tareas como eliminar columnas duplicadas o agruparlas por columnas. Para obtener más información, consulte Transformaciones personalizadas en la Guía para desarrolladores de Amazon SageMaker AI.

Generación de visualizaciones y análisis de datos

Después de importar sus datos en Data Wrangler, puede usarlos para generar visualizaciones y análisis de datos

- Visualizaciones: Data Wrangler puede generar diferentes tipos de gráficos, como histogramas y gráficos de dispersión. Por ejemplo, puede generar un histograma para identificar valores atípicos en sus datos.
- Análisis de datos: puede utilizar un Informe de información y calidad de los datos para Amazon Personalize para obtener información sobre sus datos a través de la información de datos y las estadísticas de columnas y filas. Este informe puede indicarle si tiene problemas de tipos con sus datos. Además, puede saber qué medidas puede tomar para mejorar sus datos. Estas acciones pueden ayudarlo a cumplir con los requisitos de recursos de Amazon Personalize, como los requisitos de entrenamiento de modelos, o pueden mejorar las recomendaciones.

Una vez que conozca sus datos a través de visualizaciones e información, podrá utilizar esta información como ayuda para aplicar transformaciones adicionales a fin de mejorar sus datos. O bien, si ha terminado de preparar sus datos, puede procesarlos e importarlos en Amazon Personalize. Para obtener información sobre las transformaciones de datos, consulte Transformación de datos. Para obtener información sobre el procesamiento y la importación de datos, consulte Procesamiento de los datos y su importación en Amazon Personalize.

Generación de visualizaciones

Puede usar Data Wrangler para crear diferentes tipos de gráficos, como histogramas y gráficos de dispersión. Por ejemplo, puede generar un histograma para identificar valores atípicos en sus datos. Para generar una visualización de datos, añada un paso de Análisis al flujo y, en Tipo de análisis, elija la visualización que desee crear.

Para obtener más información sobre la creación de visualizaciones en Data Wrangler, consulte Analyze and Visualize en la Guía para desarrolladores de Amazon SageMaker AI.

Generación de análisis de datos

Puede usar Data Wrangler para generar un Informe de información y calidad de datos para Amazon Personalize específico para su tipo de conjunto de datos. Antes de generar el informe, le recomendamos que transforme los datos para cumplir con los requisitos de Amazon Personalize.

Esto generará información más relevante. Para obtener más información, consulte Transformación de datos.

Temas

- Contenido del informe
- Generación del informe

Contenido del informe

El Informe de información y calidad de los datos de Amazon Personalize incluye las siguientes secciones:

- Resumen: el resumen del informe incluye estadísticas del conjunto de datos y advertencias de alta prioridad:
  - Estadísticas de conjuntos de datos: incluyen estadísticas específicas de Amazon Personalize, como el número de usuarios únicos en sus datos de interacciones, y estadísticas generales, como el número de valores atípicos o valores que faltan.
  - Advertencias de alta prioridad: se trata de información específica de Amazon Personalize que tiene el mayor impacto en el entrenamiento o las recomendaciones. Cada advertencia incluye una acción recomendada que puede realizar para resolver el problema.
- Filas duplicadas y filas incompletas: estas secciones incluyen información sobre qué filas tienen valores faltantes y qué filas están duplicadas en sus datos.
- Resumen de características: en esta sección se incluye el tipo de datos para cada columna, la información de los datos que faltan o no son válidos, y los recuentos de advertencias.
- Detalles de características: esta sección incluye subsecciones con información detallada para cada una de las columnas de datos. Cada subsección incluye estadísticas para la columna, como el recuento de valores categóricos y la información sobre los valores que faltan. Además, cada subsección incluye información específica de Amazon Personalize y acciones recomendadas para las columnas de datos. Por ejemplo, una información podría indicar que una columna tiene más de 30 categorías posibles.

Problemas de tipos de datos

El informe identifica columnas que no son del tipo de datos correcto y especifica el tipo requerido. Para obtener información relacionada con estas características, debe convertir el tipo de datos de la

columna y volver a generar el informe. Para convertir el tipo, puede utilizar la transformación de Data Wrangler Analizar valor como tipo.

Información sobre Amazon Personalize

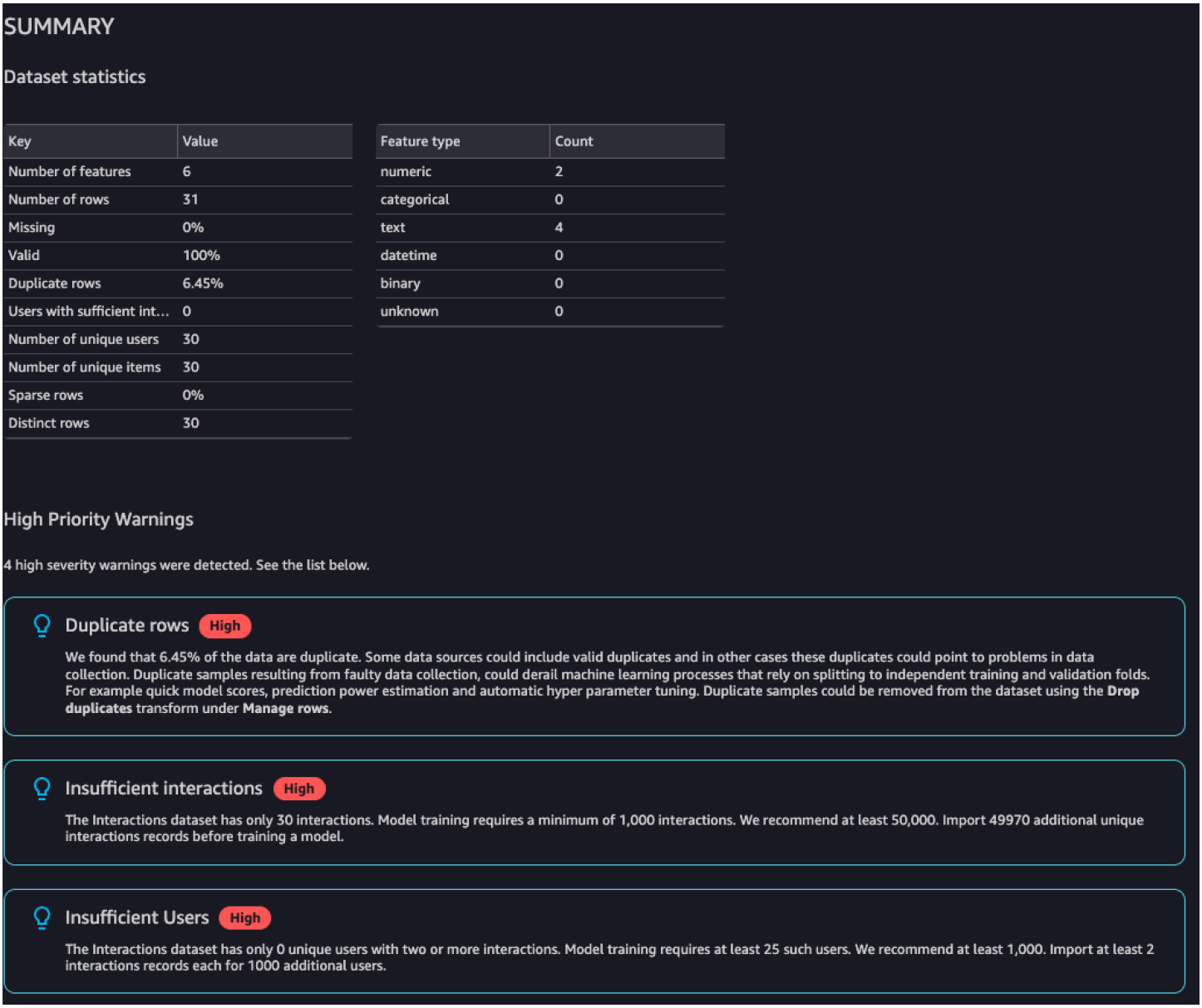
La información de Amazon Personalize incluyen un resultado y una acción sugerida. La acción es opcional. Por ejemplo, el informe puede incluir información y una acción relacionadas con el número de categorías para una columna de datos categóricos. Si no cree que la columna sea categórica, puede ignorar esta información y no realizar ninguna acción.

Salvo pequeñas diferencias de redacción, la información específica de Amazon Personalize es la misma que la información de un conjunto de datos único que puede generar al analizar sus datos con Amazon Personalize. Por ejemplo, el informe de conclusiones de Data Wrangler incluye información como “El conjunto de datos de interacciones de elementos tiene solo X usuarios únicos con dos o más interacciones”. Sin embargo, no incluye información como “El X % de los elementos del conjunto de datos de elementos no tiene interacciones en el conjunto de datos de interacciones de elementos”.

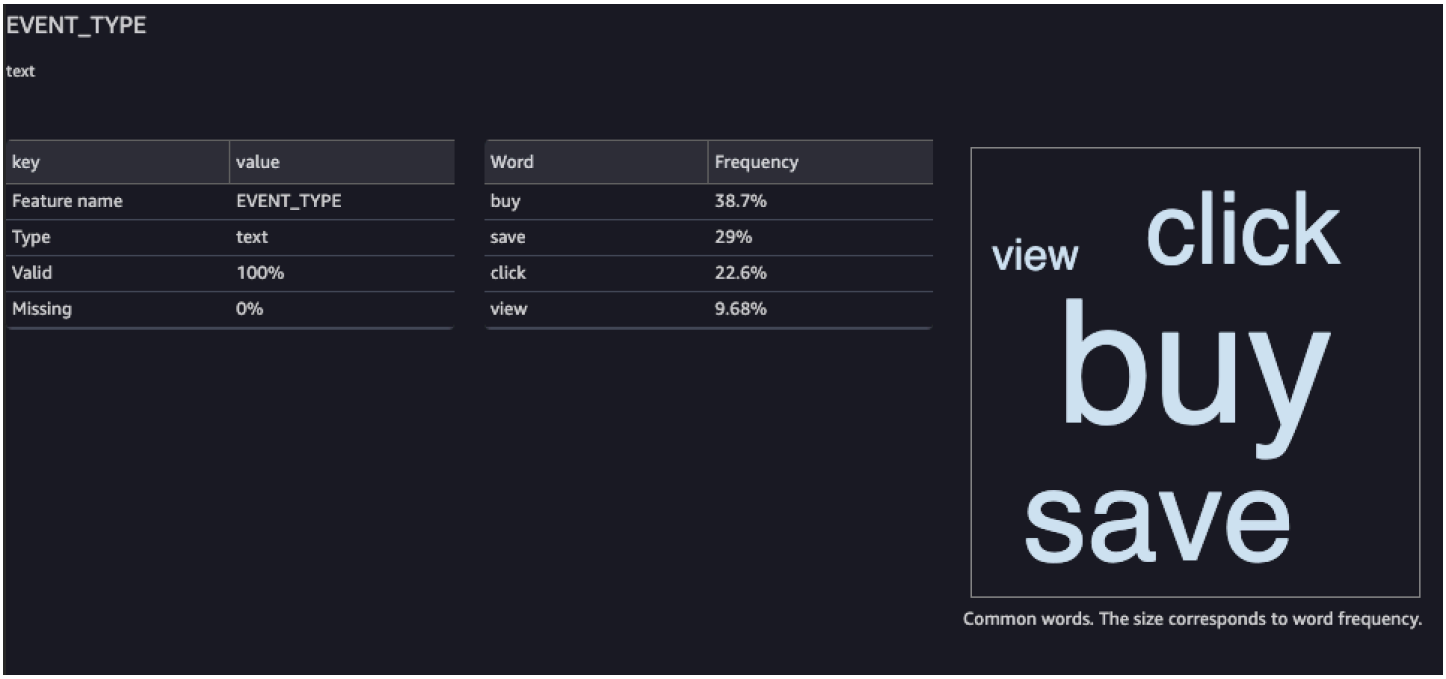
Para ver una lista de posibles conclusiones específicas de Amazon Personalize, consulte la información que no hace referencia a varios conjuntos de datos en Información sobre los datos.

Ejemplos de informes

El aspecto del informe de Amazon Personalize es el mismo que el del informe de información general de Data Wrangler. Para ver ejemplos del informe de información general, consulte Obtenga información sobre los datos y la calidad de los datos en la Guía para desarrolladores de Amazon SageMaker AI. En el siguiente ejemplo se muestra cómo funciona la sección de resumen de un informe para un conjunto de datos de interacciones de elementos. Incluye estadísticas de conjuntos de datos y algunas posibles advertencias del conjunto de datos de interacciones de elementos de alta prioridad.



El siguiente ejemplo muestra cómo puede aparecer en un informe la sección de detalles de características de una columna EVENT_TYPE de un conjunto de datos de interacciones de elementos.



Generación del informe

Para generar el Informe de información y calidad de los datos para Amazon Personalize, elija Obtener información sobre los datos para su transformación y cree un análisis.

Para generar un Informe de información y calidad de los datos para Amazon Personalize

1. Elija la opción + para la transformación que está analizando. Si no ha agregado ninguna transformación, elija el signo + para la transformación de Tipos de datos. Data Wrangler añade esta transformación automáticamente a su flujo.
2. Elija Obtener información sobre los datos. Aparece el panel Crear análisis.
3. En Tipo de análisis, elija Informe de información y calidad de los datos para Amazon Personalize.
4. En Tipo de conjunto de datos, elija el tipo de conjunto de datos de Amazon Personalize que está analizando.
5. Si lo desea, elija Ejecutar en datos completos. De forma predeterminada, Data Wrangler genera información sobre solo una muestra de sus datos.
6. Seleccione Crear. Cuando se complete el análisis, aparecerá el informe.

Procesamiento de los datos y su importación en Amazon Personalize

Cuando haya terminado de analizar y transformar sus datos, estará listo para procesarlos e importarlos en Amazon Personalize.

- Procesamiento de datos: el procesamiento de los datos aplica la transformación a todo el conjunto de datos y la envía al destino que especifique. En este caso, especifique un bucket de Amazon S3.
- Importación de datos a Amazon Personalize: para importar datos procesados a Amazon Personalize, ejecute un Jupyter Notebook incluido en SageMaker AI Studio Classic. Este cuaderno crea sus conjuntos de datos de Amazon Personalize e importa sus datos en ellos.

Procesamiento de datos

Antes de importar datos en Amazon Personalize, debe aplicar la transformación a todo el conjunto de datos y enviarlos a un bucket de Amazon S3. Para ello, debe crear un nodo de destino con el destino establecido en un bucket de Amazon S3 y, a continuación, lanzar un trabajo de procesamiento para la transformación.

Para obtener instrucciones paso a paso sobre cómo especificar un destino y lanzar un trabajo de proceso, consulte Iniciar trabajos de procesamiento con unos pocos clics con Amazon SageMaker AI Data Wrangler. Cuando añada un destino, elija Amazon S3. Utilizará esta ubicación al importar los datos procesados en Amazon Personalize.

Cuando termine de procesar los datos, estará listo para importarlos del bucket de Amazon S3 a Amazon Personalize.

Importación de datos en Amazon Personalize

Tras procesar los datos, estará listo para importarlos en Amazon Personalize. Para importar los datos procesados a Amazon Personalize, ejecute un Jupyter Notebook incluido en SageMaker AI Studio Classic. Este cuaderno crea sus conjuntos de datos de Amazon Personalize e importa sus datos en ellos.

Para importar datos procesados en Amazon Personalize

1. Para la transformación que desee exportar, elija Exportar a y Amazon Personalize (mediante el cuaderno de Jupyter).
2. Modifique el cuaderno para especificar el bucket de Amazon S3 que utilizó como destino de los datos para el trabajo de procesamiento. Si lo desea, especifique el dominio para su grupo de

conjuntos de datos. De manera predeterminada, el cuaderno crea un grupo de conjuntos de datos personalizados.

3. Revise las celdas del cuaderno que crean el esquema. Compruebe que los campos del esquema tienen los tipos y atributos esperados antes de ejecutar la celda.

   - Compruebe que los campos que admiten datos nulos incluyen `null` en la lista de tipos. En el ejemplo siguiente se muestra cómo añadir `null` para un campo.

     ```
     {
       "name": "GENDER",
       "type": [
         "null",
         "string"
       ],
       "categorical": true
     }
     ```

   - Compruebe que los campos categóricos tengan el atributo categórico establecido en verdadero. En el siguiente ejemplo se muestra cómo marcar un campo categórico.

     ```
     {
               "name": "SUBSCRIPTION_MODEL",
               "type": "string",
               "categorical": true
     }
     ```

   - Compruebe que los campos textuales tengan el atributo textual establecido en verdadero. En el siguiente ejemplo se muestra cómo marcar un campo como textual.

     ```
     {
           "name": "DESCRIPTION",
           "type": [
             "null",
             "string"
           ],
           "textual": true
     }
     ```

4. Ejecute el cuaderno para crear un esquema y un conjunto de datos, e importe los datos en el conjunto de datos de Amazon Personalize. Utiliza el bloc de notas del mismo modo que lo haría con un bloc de notas fuera de SageMaker AI Studio Classic. Para obtener información sobre

cómo ejecutar los cuadernos de Jupyter, consulte Ejecución de código. Para obtener información sobre las libretas de SageMaker AI Studio Classic, consulta Cómo usar las libretas Amazon SageMaker AI en la Guía para desarrolladores de Amazon SageMaker AI.

Después de completar el cuaderno, si ha importado datos de interacciones, estará listo para crear generadores de recomendaciones o recursos personalizados. O puede repetir el proceso con un conjunto de datos de elementos o un conjunto de datos de usuarios.

- Para obtener información sobre cómo crear recomendadores de dominios, consulte Recomendadores de dominios en Amazon Personalize.
- Para obtener información sobre cómo crear e implementar recursos personalizados, consulte Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize.

Importación de registros individuales a un conjunto de datos de Amazon Personalize

Una vez finalizado Creación de un esquema y un conjunto de datos, puede importar registros individuales en un conjunto de datos existente, incluidas las interacciones de elementos, usuarios, elementos, acciones o interacciones de acción. La importación de datos de forma individual le permite añadir pequeños lotes de registros a sus conjuntos de datos de Amazon Personalize a medida que su catálogo crece. Puede importar hasta 10 registros por operación de importación individual.

Si importa un artículo, usuario o acción con el mismo ID que otro registro que ya se encuentra en su conjunto de datos, Amazon Personalize lo sustituye por el nuevo registro. Si registra dos eventos de interacción con artículos o de interacción de acción con la misma marca de tiempo e idénticas propiedades, Amazon Personalize solo conserva uno de ellos.

Si usa Apache Kafka, puede usar el conector de Kafka para Amazon Personalize para transmitir datos en tiempo real a Amazon Personalize. Para obtener más información, consulte Conector de Kafka para Amazon Personalize en el repositorio de Github personalize-kafka-connector.

Si tiene una gran cantidad de registros históricos, le recomendamos que primero importe los datos de forma masiva y, a continuación, los importe individualmente según sea necesario. Consulte Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos.

Filtre las actualizaciones para importaciones de registros individuales

Amazon Personalize actualiza los filtros que haya creado en el grupo de conjuntos de datos con sus nuevos datos de interacciones, elementos y usuarios en un plazo de 20 minutos a partir de la última importación individual. Esta actualización permite que sus campañas usen sus datos más recientes al filtrar las recomendaciones para sus usuarios.

Si ya creó un generador de recomendaciones o implementó una versión de solución personalizada con una campaña, la forma en que los nuevos registros individuales influyen en las recomendaciones depende del caso de uso del dominio o de la receta que utilice. Para obtener más información, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Temas

- Importación de interacciones de forma individual
- Importación de usuarios de forma individual
- Importación de elementos de forma individual
- Importación de acciones de forma individual

Importación de interacciones de forma individual

Una vez que haya completado Creación de un esquema y un conjunto de datos para crear un conjunto de datos de interacciones de elementos, puede importar individualmente uno o más eventos nuevos en el conjunto de datos. Para importar los eventos de interacciones de forma individual, cree un rastreador de eventos y, a continuación, importe uno o más eventos en su conjunto de datos de interacciones de elementos. Puede importar eventos históricos de interacciones individuales mediante la consola Amazon Personalize, o importar eventos históricos o en tiempo real con AWS Command Line Interface (AWS CLI) o AWS SDKs.

En esta sección, se incluye información sobre la importación de eventos con la consola de Amazon Personalize. Recomendamos usar la consola de Amazon Personalize para importar únicamente eventos históricos. Para obtener información sobre el uso de AWS CLI o AWS SDKs para registrar eventos en tiempo real, consulteRegistro de los eventos en tiempo real para influir en las recomendaciones.

Para obtener información sobre cómo Amazon Personalize actualiza los filtros de los nuevos registros y cómo los nuevos registros influyen en las recomendaciones, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize.

Temas

- Creación de un rastreador de eventos (consola)
- Importación de eventos de forma individual (consola)

Creación de un rastreador de eventos (consola)

> **Note**
>
> Si ha creado un rastreador de eventos, puede pasar a Importación de eventos de forma individual (consola).

Para poder importar un evento en un conjunto de datos de interacciones, debe crear un rastreador de eventos para el grupo de conjuntos de datos.

Para crear un rastreador de eventos (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con el conjunto de datos de interacciones de elementos en el que quiera importar los eventos.
3. En el Panel para el grupo de conjuntos de datos, en Instalar el SDK de ingesta de eventos, elija Iniciar.
4. En la página Configurar rastreador, en Configuraciones del rastreador, para Nombre del rastreador, proporcione un nombre para el rastreador de eventos y elija Siguiente.
5. La página Instalar el SDK muestra el ID de seguimiento del nuevo rastreador de eventos e instrucciones para usar AWS Amplify o AWS Lambda transmitir los datos de los eventos.

   Puede ignorar esta información porque está utilizando la consola de Amazon Personalize para subir datos de eventos. Si desea transmitir datos de eventos utilizando AWS Amplify o AWS Lambda en el futuro, puede ver esta información eligiendo el rastreador de eventos en la página de rastreadores de eventos.
6. Seleccione Finalizar. Ahora puede importar eventos con la consola (consulte Importación de eventos de forma individual (consola) o registrar eventos en tiempo real mediante la operación `PutEvents`, consulte Registro de los eventos en tiempo real para influir en las recomendaciones).

Importación de eventos de forma individual (consola)

Después de crear un rastreador de eventos, puede importar los eventos de forma individual en un conjunto de datos de interacciones de elementos. En este procedimiento, se supone que ya ha creado un conjunto de datos de interacciones de elementos. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Para importar eventos de forma individual (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con el conjunto de datos de interacciones de elementos en el que quiera importar los eventos.
3. Elija conjuntos de datos en el panel de navegación.
4. En la página Conjuntos de datos, elija el conjunto de datos de interacciones.
5. En la parte superior derecha de la página de detalles del conjunto de datos, elija Modificar conjunto de datos y Crear registro.
6. En la página Crear registros de interacciones entre usuarios y elementos, en Entrada de registro, introduzca los detalles del evento en formato JSON. Los nombres y los valores de los campos del evento deben coincidir con el esquema que utilizó al crear el conjunto de datos de interacciones de elementos. Amazon Personalize proporciona una plantilla JSON con nombres de campos y tipos de datos de este esquema. Puede importar hasta 10 eventos a la vez.
7. Elija Crear registros. En Respuesta, se muestra el resultado de la importación y un mensaje de éxito o error.

Importación de usuarios de forma individual

Una vez que haya terminado Creación de un esquema y un conjunto de datos para crear un conjunto de datos de usuarios, puede importar individualmente uno o más usuarios nuevos en el conjunto de datos. La importación individual de usuarios le permite mantener actualizado su conjunto de datos de usuarios con importaciones de lotes pequeños a medida que su catálogo crece. Puede importar hasta 10 usuarios a la vez. Si tiene una gran cantidad de usuarios nuevos, le recomendamos que primero importe los datos de forma masiva y, a continuación, los importe individualmente según sea necesario. Consulte Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos.

Puedes usar la consola Amazon Personalize, la AWS Command Line Interface (AWS CLI) o AWS SDKs para importar usuarios. Si importa un usuario con el mismo `userId` que un usuario que ya se encuentra en su conjunto de datos de usuarios, Amazon Personalize lo sustituye por el nuevo elemento. Puede importar hasta 10 usuarios a la vez.

Para obtener información sobre cómo Amazon Personalize actualiza los filtros de los nuevos registros y cómo los nuevos registros influyen en las recomendaciones, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize.

Temas

- Importación de usuarios de forma individual (consola)
- Importación de usuarios de forma individual (AWS CLI)
- Importación de usuarios de forma individual (AWS SDKs)

Importación de usuarios de forma individual (consola)

Puede importar hasta 10 usuarios a la vez. En este procedimiento, se presupone que ya ha creado un conjunto de datos de usuarios. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Para importar usuarios de forma individual (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con el conjunto de datos de usuarios al que desee importar el usuario.
3. Elija Conjuntos de datos en el panel de navegación.
4. En la página Conjuntos de datos, elija el conjunto de datos de usuarios.
5. En la página de detalles del conjunto de datos, en la parte superior derecha, elija Modificar conjunto de datos y Crear registro.
6. En la página Crear registros de usuario, para la entrada de registro, introduzca los detalles del usuario en formato JSON. Los nombres y valores de los campos del usuario deben coincidir con el esquema que utilizó al crear el conjunto de datos de usuarios. Amazon Personalize proporciona una plantilla JSON con nombres de campos y tipos de datos de este esquema.
7. Elija Crear registros. En Respuesta, se muestra el resultado de la importación y un mensaje de éxito o error.

Importación de usuarios de forma individual (AWS CLI)

Agregue uno o más usuarios a su conjunto de datos de usuarios mediante la operación PutUsers. Puede importar hasta 10 usuarios con una sola llamada `PutUsers`. En esta sección se supone que ya ha creado un conjunto de datos de usuarios. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Use el siguiente comando `put-users` para añadir uno o más usuarios con la AWS CLI. Sustituya `dataset arn` por el nombre de recurso de Amazon (ARN) de su conjunto de datos y `user Id` por el ID del usuario. Si un usuario con el mismo `userId` ya se encuentra en su conjunto de datos de usuarios, Amazon Personalize lo sustituye por el nuevo.

Para `properties`, para cada campo de su conjunto de datos de usuarios, sustituya `propertyName` por el nombre del campo de su esquema en formato camel case. Por ejemplo, GENDER sería `gender` y MEMBERSHIP_TYPE sería `membershipType`. Sustituya `user data` por los datos del usuario. En el caso de los datos de cadena categóricos, para incluir varias categorías para un sola propiedad, separe cada categoría con un separador vertical (`|`). Por ejemplo, `\"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\"}"
    }]'
```

Importación de usuarios de forma individual (AWS SDKs)

Agregue uno o más usuarios a su conjunto de datos de usuarios mediante la operación PutUsers. Si un usuario con el mismo `userId` ya se encuentra en su conjunto de datos de usuarios, Amazon Personalize lo sustituye por el nuevo. Puede importar hasta 10 usuarios con una sola llamada `PutUsers`. En esta sección se supone que ya ha creado un conjunto de datos de usuarios. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

El siguiente código muestra cómo añadir uno o más usuarios a su conjunto de datos de usuarios. Para cada parámetro de nombre de propiedad, pase el nombre del campo de su esquema en formato camel case. Por ejemplo, GENDER sería `gender` y MEMBERSHIP_TYPE sería `membershipType`. Para cada parámetro de valor de propiedad, pase los datos al usuario.

En el caso de los datos de cadena categóricos, para incluir varias categorías para un sola propiedad, separe cada categoría con un separador vertical (`|`). Por ejemplo, `"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();
```

Importación de elementos de forma individual

Una vez que haya terminado Creación de un esquema y un conjunto de datos para crear un conjunto de datos de elementos, puede importar individualmente uno o más elementos nuevos en el conjunto de datos. La importación individual de elementos le permite mantener actualizado su conjunto de datos de elementos con importaciones de lotes pequeños a medida que su catálogo crece. Puede importar hasta 10 elementos a la vez. Si tiene una gran cantidad de elementos nuevos, le recomendamos que primero importe los datos de forma masiva y, a continuación, los importe individualmente según sea necesario. Consulte Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos.

Puedes usar la consola Amazon Personalize, la AWS Command Line Interface (AWS CLI) o AWS SDKs para importar artículos. Si importa un elemento con el mismo `itemId` que un elemento que ya

se encuentra en su conjunto de datos de elementos, Amazon Personalize lo sustituye por el nuevo elemento.

Para obtener información sobre cómo Amazon Personalize actualiza los filtros de los nuevos registros y cómo los nuevos registros influyen en las recomendaciones, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize.

Temas

- Importación de elementos de forma individual (consola)
- Importación de elementos de forma individual (AWS CLI)
- Importación de elementos de forma individual (AWS SDKs)

Importación de elementos de forma individual (consola)

Puede importar hasta 10 elementos en un conjunto de datos de elementos a la vez. En este procedimiento, se asume que ya ha creado un conjunto de datos de elementos. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Para importar elementos de forma individual (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con el conjunto de datos de elementos al que desee importar los elementos.
3. Elija Conjuntos de datos en el panel de navegación.
4. En la página Conjuntos de datos, elija el conjunto de datos de elementos.
5. En la parte superior derecha de la página de detalles del conjunto de datos, elija Modificar conjunto de datos y, a continuación, Crear registro.
6. En la página Crear registros de elementos, en Entrada de registro, introduzca los detalles del elemento en formato JSON. Los nombres y valores de los campos del elemento deben coincidir con el esquema que utilizó al crear el conjunto de datos de elementos. Amazon Personalize proporciona una plantilla JSON con nombres de campos y tipos de datos de este esquema.
7. Elija Crear registros. En Respuesta, se muestra el resultado de la importación y un mensaje de éxito o error.

Importación de elementos de forma individual (AWS CLI)

Agregue uno o más elementos a su conjunto de datos de elementos mediante la operación PutItems. Puede importar hasta 10 elementos con una sola llamada `PutItems`. En esta sección, se asume que ya ha creado un conjunto de datos de elementos. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Use el siguiente comando `put-items` para agregar uno o más elementos con la AWS CLI. Sustituya `dataset arn` por el nombre de recurso de Amazon (ARN) de su conjunto de datos y `item Id` por el ID del elemento. Si un elemento con el mismo `itemId` ya se encuentra en su conjunto de datos de elementos, Amazon Personalize lo sustituye por el nuevo.

Para `properties`, para cada campo de su conjunto de datos de elementos, sustituya `propertyName` por el nombre del campo de su esquema en formato camel case. Por ejemplo, GENRES sería `genres` y CREATION_TIMESTAMP sería creationTimestamp. Sustituya los `item data` por los datos del elemento. Los datos de `CREATION_TIMESTAMP` deben estar en formato de tiempo Unix y en segundos. En el caso de los datos de cadena categóricos, para incluir varias categorías para un sola propiedad, separe cada categoría con un separador vertical (`|`). Por ejemplo, `\"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\"}"
    }]'
```

Importación de elementos de forma individual (AWS SDKs)

Agregue uno o más elementos a su conjunto de datos de elementos mediante la operación PutItems. Puede importar hasta 10 elementos con una sola llamada `PutItems`. Si un elemento con el mismo `itemId` ya se encuentra en su conjunto de datos de elementos, Amazon Personalize lo sustituye por el nuevo. En esta sección, se asume que ya ha creado un conjunto de datos de elementos. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

El siguiente código muestra cómo añadir uno o más elementos a su conjunto de datos de elementos. Para cada parámetro de nombre de propiedad, pase el nombre del campo de su esquema en formato camel case. Por ejemplo, GENRES sería `genres` y CREATION_TIMESTAMP sería `creationTimestamp`. Para cada parámetro de valor de la propiedad, pase los datos para el elemento. Los datos de `CREATION_TIMESTAMP` deben estar en formato de tiempo Unix y en segundos.

En el caso de los datos de cadena categóricos, para incluir varias categorías para un sola propiedad, separe cada categoría con un separador vertical (`|`). Por ejemplo, `"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();
```

Importación de acciones de forma individual

Una vez que haya completado Creación de un esquema y un conjunto de datos para crear un conjunto de datos de acciones, puede importar una o más acciones nuevas de forma individual en el conjunto de datos. Al importar acciones individualmente, puede mantener actualizado su conjunto de datos de acciones con importaciones de lotes pequeños a medida que su catálogo crece. Puede importar hasta 10 acciones a la vez. Si tiene una gran cantidad de acciones nuevas, le recomendamos que primero importe los datos de forma masiva y, a continuación, importe los datos

de las acciones de forma individual según sea necesario. Consulte Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos.

Puedes usar la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) o AWS SDKs para importar acciones. Si importa una acción con el mismo `actionId` que otra acción que ya se encuentra en su conjunto de datos de acciones, Amazon Personalize la sustituye por la nueva acción.

Para obtener información sobre cómo los nuevos registros influyen en las recomendaciones, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Temas

- Importación de acciones de forma individual (consola)
- Importación de acciones de forma individual (AWS CLI)
- Importación de acciones de forma individual (AWS SDKs)

Importación de acciones de forma individual (consola)

Puede importar hasta 10 acciones a la vez en un conjunto de datos de acciones. En esta sección se supone que ya ha creado un conjunto de datos de acciones. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Para importar acciones de forma individual (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con el conjunto de datos de acciones en el que quiera realizar adiciones.
3. En el panel de navegación de la izquierda, elija Conjuntos de datos.
4. En la página Conjuntos de datos, elija el conjunto de datos de acciones.
5. En la parte superior derecha de la página de detalles del conjunto de datos, elija Modificar conjunto de datos y, a continuación, Crear registro.
6. En la página Crear registros de acciones, en Entrada de registro, introduzca los detalles de la acción en formato JSON. Los nombres y los valores de los campos de la acción deben coincidir con el esquema que utilizó al crear el conjunto de datos de acciones. Amazon Personalize proporciona una plantilla JSON con nombres de campos y tipos de datos de este esquema.

7. Elija Crear registros. En Respuesta, se muestra el resultado de la importación y un mensaje de éxito o error.

Importación de acciones de forma individual (AWS CLI)

Agregue una o varias acciones al conjunto de datos de acciones mediante la operación de la API `PutActions`. Puede importar hasta 10 acciones a la vez. En esta sección se supone que ya ha creado un conjunto de datos de acciones. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

Use el siguiente comando `put-actions` para agregar una o más acciones con la AWS CLI. Sustituya `dataset arn` por el nombre de recurso de Amazon (ARN) de su conjunto de datos y `actionId` por el ID de la acción. Si una acción con el mismo valor `actionId` ya se encuentra en su conjunto de datos de acciones, Amazon Personalize lo sustituye por el nuevo.

Para `properties`, en cada campo del conjunto de datos de acciones, sustituya `propertyName` por el nombre del campo de su esquema en formato camel case. Por ejemplo, ACTION_EXPIRATION_TIMESTAMP sería `actionExpirationTimestamp` y CREATION_TIMESTAMP sería creationTimestamp. Sustituya `property data` por los datos de la propiedad.

```
aws personalize-events put-actions \
  --dataset-arn dataset arn \
  --actions '[{
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    },
    {
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    }]'
```

Importación de acciones de forma individual (AWS SDKs)

Añada una o más acciones a su conjunto de datos de acciones mediante la PutActions operación. Puede importar hasta 10 acciones con una sola llamada a `PutActions`. Si una acción con el mismo valor `actionId` ya se encuentra en su conjunto de datos de acciones, Amazon Personalize lo sustituye por el nuevo. En esta sección se supone que ya ha creado un conjunto de datos de

acciones. Para obtener información sobre cómo crear conjuntos de datos, consulte Creación de un esquema y un conjunto de datos.

El siguiente código muestra cómo agregar una o más acciones al conjunto de datos de acciones. Para cada acción, especifique el valor `actionId`. Si una acción con el mismo valor `actionId` ya se encuentra en su conjunto de datos de acciones, Amazon Personalize lo sustituye por el nuevo. Para `properties`, en cada campo adicional del conjunto de datos de acciones, sustituya `propertyName` por el nombre del campo de su esquema en formato camel case. Por ejemplo, ACTION_EXPIRATION_TIMESTAMP sería `actionExpirationTimestamp` y CREATION_TIMESTAMP sería creationTimestamp. Sustituya `property data` por los datos de la propiedad.

```
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\"}"
      }]
)
```

Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize

Después de importar datos en un conjunto de datos de interacciones de elementos, usuarios o elementos, puede usar la consola de Amazon Personalize para analizar los datos. Puede obtener información sobre sus datos a través de la información de los datos y las estadísticas de columnas y filas. Además, puede saber qué medidas puede tomar para mejorar sus datos. Estas acciones pueden ayudarlo a cumplir con los requisitos de recursos de Amazon Personalize, como los requisitos de entrenamiento de modelos, o pueden mejorar las recomendaciones.

> **Important**
>
> No se puede usar la consola Amazon Personalize para analizar los datos de un conjunto de datos de interacciones de acción o acciones.

Tras realizar los cambios recomendados, puede volver a importar los datos y comprobar si ha resuelto algún problema o ha mejorado las estadísticas del conjunto de datos. Para obtener información sobre la actualización de datos, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Si no ve ninguna información, sus datos se ajustan a las expectativas de datos de Amazon Personalize. Puede analizar los datos de un grupo de conjuntos de datos de dominio o de un grupo de conjuntos de datos personalizados.

Al generar información y calcular estadísticas, Amazon Personalize tiene en cuenta todos los datos masivos y transmitidos de usuarios no anónimos. Los eventos de usuarios anónimos no se tienen en cuenta hasta que los asocie a un `userId`. Para obtener más información, consulte Registro de eventos para usuarios anónimos.

Temas

- Permisos necesarios para analizar datos
- Información sobre los datos
- Visualización de estadísticas e información del conjunto de datos

Permisos necesarios para analizar datos

Si concede a los usuarios acceso completo a Amazon Personalize, no será necesario cambiar los permisos. Si concede a sus usuarios solo los permisos necesarios para realizar una tarea en Amazon Personalize, su política AWS Identity and Access Management (IAM) debe incluir las siguientes acciones adicionales de información de datos.

- personalizar: CreateDataInsightsJob
- personalizar: ListDataInsightsJobs
- personalizar: DescribeDataInsightsJob
- personalizar: GetDataInsights

Información sobre los datos

La siguiente es la posible información de datos que puede generar en Amazon Personalize.

Información	Action	Conjuntos de datos relacionados
El conjunto de datos de interacciones solo tiene X interacciones. El entrenamiento de modelos requiere un mínimo de 1000 interacciones. Se recomiendan como mínimo 50 000.	Importe registros de interacciones únicas adicionales Y antes de entrenar un modelo.	Interacciones de elementos
El conjunto de datos de interacciones solo tiene X usuarios únicos con dos o más interacciones. El entrenamiento de modelos requiere al menos 25 de estos usuarios. Se recomiendan como mínimo 1000.	Importe al menos 2 registros de interacciones para cada uno de los usuarios adicionales Y.	Interacciones de elementos
El X % de los elementos del conjunto de datos de elementos no tienen interacci	Asegúrese de importar todos los datos de sus interacciones y compruebe si no hay	Interacciones de

Información	Action	Conjuntos de datos relacionados
ones en el conjunto de datos de interacciones, por lo que es posible que no se recomienden.	coincidencias IDs entre sus elementos y los conjuntos de datos de interacciones. Consulte las estadísticas de los conjuntos de datos a continuación para sus conjuntos de datos de interacciones y elementos para asegurarse de que ha importado el número de filas esperado. Si su caso de uso o receta utiliza la exploración, modifique la configuración de exploración para recomendar más elementos sin datos de interacciones.	elementos y elementos
El X % de los usuarios del conjunto de datos de usuarios no tienen interacciones en el conjunto de datos de interacciones. Estos usuarios recibirán recomendaciones de elementos populares.	Asegúrese de importar todos los datos de sus interacciones y compruebe si no hay coincidencias IDs entre sus conjuntos de datos de usuarios y de interacciones. Consulte las estadísticas de los conjuntos de datos a continuación para sus usuarios y conjuntos de datos de interacciones para asegurarse de que ha importado la cantidad de filas esperada. Importe cualquier interacción adicional para que más usuarios tengan datos de interacciones.	Interacciones de elementos y usuarios

Información	Action	Conjuntos de datos relacionados
El conjunto de datos <Users or Items or Interactions> tiene un X % de filas en las que falta un valor. Esto puede afectar negativamente a las recomendaciones. Recomendamos que todos los campos obligatorios y opcionales estén completados al menos en un 70 %.	Importe registros completos adicionales o vuelva a importar datos sin filas incompletas, o a importar datos de nuevo en los que falten valores y los sustituya por otros datos, como el promedio de las columnas numéricas o el valor más común de las columnas categóricas.	Cualquiera
Las siguientes columnas del <datasetType>conjunto de datos están completas en menos del 70%: <ColumnName, ColumnName... >. Si estos datos se incluyen en el entrenamiento, las recomendaciones se pueden ver afectadas negativamente. Recomendamos que las columnas que permiten valores nulos estén completas al menos en un 70 %.	Importe registros completos adicionales o vuelva a importar datos sin filas incompletas, o a importar datos de nuevo en los que falten valores y los sustituya por otros datos, como el promedio de las columnas numéricas o el valor más común de las columnas categóricas.	Cualquiera
Las siguientes columnas (numéricas) tienen valores atípicos: <ColumnName, ColumnName... >. Los valores atípicos no son siempre un problema, pero a veces afectan negativamente a las recomendaciones.	Con las Estadísticas de columna que aparecen a continuación, compruebe si los valores mínimo y máximo de estas columnas coinciden con sus expectativas. Si estos valores son inesperados, compruebe si hay imprecisiones en los datos de estas columnas, y revise la recopilación y el procesamiento de datos para ver si hay problemas.	Cualquiera

Información	Action	Conjuntos de datos relacionados
Las siguientes columnas tienen más de 1000 categorías posibles: <ColumnName, ColumnName... >. Si estos datos se incluyen en la formación, pueden afectar negativamente a las recomendaciones: <ColumnName, ColumnName... >.	Revise sus datos categóricos para ver si hay problemas, como categorías duplicadas causadas por variaciones en la ortografía. Resuelva cualquier imprecisión y vuelva a importar los datos.	Cualquiera
Las siguientes columnas de metadatos textuales están completas en menos del 85% y no se utilizarán en el entrenamiento con modelos: <ColumnName, ColumnName... >.	Importe filas adicionales o vuelva a importar las filas con datos de texto para estas columnas.	Items
El conjunto de datos de interacciones tiene más de 10 tipos de eventos únicos, lo que provocará un error en el entrenamiento de modelos.	Revisa la columna de tipos de eventos para ver si hay imprecisiones, como tipos de eventos duplicados causados por variaciones ortográficas. Elimine los tipos de eventos innecesarios y vuelva a importar los datos.	Interacciones de elementos
El conjunto de datos de interacciones tiene la misma marca de tiempo para todos los registros. Si usa una receta USER_SEGMENTATION y todos los registros tienen la misma marca de tiempo, se producirá un error en el entrenamiento del modelo.	Compruebe sus datos para ver si hay problemas con las marcas de tiempo y reemplace las marcas de tiempo duplicadas por marcas de tiempo únicas.	Interacciones de elementos

Visualización de estadísticas e información del conjunto de datos

Para ver información y estadísticas sobre sus datos en los conjuntos de datos de Amazon Personalize, navegue hasta los conjuntos de datos en la consola de Amazon Personalize y elija la opción de ejecutar análisis.

> **Note**
>
> Para que Amazon Personalize pueda analizar los datos, debe importarlos antes. Para obtener más información, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize

Para ver información y estadísticas

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En el panel de navegación, elija Análisis de datos en Conjuntos de datos.
4. En la parte superior derecha, elija Ejecutar análisis. Amazon Personalize comienza a analizar sus datos. Este proceso puede tardar hasta 15 minutos. Si se realiza correctamente, los resultados aparecen en esta página.
5. En Información, utilice lo siguiente para filtrar la información que aparece.
   - Para encontrar información que incluya un lenguaje específico, introduzca sus criterios en Buscar información. A medida que escriba texto, la lista se actualizará para incluir solo la información con la cadena exacta en la información o la acción recomendada.
   - Para filtrar la información por tipo de conjunto de datos, cambie Todos los conjuntos de datos por el tipo de conjunto de datos específico. La lista se actualiza para incluir solo información relacionada con este conjunto de datos.
6. Para ver las estadísticas de un conjunto de datos, haga lo siguiente.
   - Para ver los detalles generales y las estadísticas de un conjunto de datos, como el número de filas, los usuarios únicos y los elementos únicos de un conjunto de datos de interacciones, amplíe la sección correspondiente al conjunto de datos.
   - Para ver las estadísticas detalladas de una columna, amplíe la sección del conjunto de datos, elija Estadísticas a nivel de columna y pulse el botón de opción de la columna.

7. Corrija cualquier problema en los datos, impórtelos de nuevo y ejecute otro análisis para verificarlos. Para obtener más información sobre la importación de datos, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Recomendadores de dominios en Amazon Personalize

Si ha creado un grupo de conjuntos de datos de dominio. después de importar los datos, podrá crear recomendadores para los casos de uso de dominio. Un generador de recomendaciones es un recurso de grupo de conjuntos de datos de dominio que genera recomendaciones. Utiliza un generador de recomendaciones en su aplicación para obtener recomendaciones en tiempo real sobre la operación GetRecommendations.

Cuando crea un recomendador, especifica un caso de uso y Amazon Personalize entrena los modelos que lo respaldan con las mejores configuraciones para cada caso de uso. Cada caso de uso tiene diferentes requisitos de API para obtener recomendaciones. Para obtener una lista de los casos de uso de generadores de recomendaciones por dominio, consulte Elección de un caso de uso. Puede crear un máximo de 15 generadores de recomendaciones por región.

Amazon Personalize vuelve a entrenar automáticamente los modelos que respaldan sus generadores de recomendaciones cada 7 días. Se trata de un reentrenamiento completo que crea modelos completamente nuevos basados en la totalidad de los datos de sus conjuntos de datos. Con los casos de uso Mejores opciones y Recomendaciones, Amazon Personalize actualiza los modelos existentes cada dos horas para incluir nuevos elementos en las recomendaciones con la exploración.

Al crear un recomendador, puede hacer lo siguiente:

- Habilitar los metadatos de elemento en las recomendaciones. Para obtener más información, consulte Habilitación de los metadatos en las recomendaciones.
- Configurar las columnas utilizadas en el entrenamiento de los modelos que respaldan el recomendador. Para obtener más información, consulte Configuración de las columnas utilizadas al crear un recomendador de dominios de Amazon Personalize.
- Para los casos de uso Mejores opciones y Recomendaciones para usted, puede configurar la exploración. Para obtener más información, consulte Configuración de la exploración de un recomendador de dominios.

Después de crear un recomendador, podrá hacer lo siguiente:

- Evaluar el recomendador: puede evaluar el rendimiento del recomendador mediante métricas en línea y sin conexión. Para obtener más información, consulte Evaluación de un recomendador de dominios de Amazon Personalize.

- Detener e iniciar el recomendador: si desea pausar la facturación de un recomendador activo, puede detenerlo y reiniciarlo más adelante. Para obtener más información, consulte Detención de un recomendador.
- Actualizar la configuración del recomendador: puede actualizar las columnas que el recomendador utiliza en el entrenamiento y actualizar la capacidad de solicitud del recomendador. Para obtener más información, consulte Actualización de un recomendador.
- Eliminar el recomendador: puede eliminar los recomendadores con la operación DeleteRecommender. También puede eliminar un generador de recomendaciones de la página de detalles del generador de recomendaciones en la consola de Amazon Personalize.

Temas

- Estados de los generadores de recomendaciones
- Solicitudes de recomendación mínimas por segundo y escalado automático
- Creación de recomendadores de dominios en Amazon Personalize
- Habilitación de los metadatos en las recomendaciones para un recomendador de dominios en Amazon Personalize
- Configuración de las columnas utilizadas al crear un recomendador de dominios de Amazon Personalize
- Configuración de la exploración de un recomendador de dominios
- Evaluación de un recomendador de dominios de Amazon Personalize
- Actualización de un recomendador
- Detención de un recomendador

Estados de los generadores de recomendaciones

Un recomendador puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Para obtener el estado de generador de recomendaciones, vaya a la página Generadores de recomendaciones de la consola de Amazon Personalize o utilice la operación DescribeRecommender.

Solicitudes de recomendación mínimas por segundo y escalado automático

> **Important**
>
> Un valor elevado de `minRecommendationRequestsPerSecond` aumentará su factura. Le recomendamos empezar con 1 para `minRecommendationRequestsPerSecond` (el valor predeterminado). Realiza un seguimiento de tu consumo mediante CloudWatch las métricas de Amazon y auméntalo `minRecommendationRequestsPerSecond` según sea necesario.

Al crear un recomendador, puede configurar las solicitudes de recomendación mínimas por segundo de este. El número mínimo de solicitudes de recomendación por segundo (`minRecommendationRequestsPerSecond`) especifica el rendimiento básico de las solicitudes de recomendación aprovisionado por Amazon Personalize. El valor predeterminado minRecommendationRequests PerSecond es`1`. Una solicitud de recomendación es una operación `GetRecommendations` única. El rendimiento de las solicitudes se mide en solicitudes por segundo y Amazon Personalize utiliza las solicitudes por segundo para derivar las solicitudes por hora y el precio del uso del recomendador.

Si sus solicitudes por segundo aumentan más allá de `minRecommendationRequestsPerSecond`, Amazon Personalize escala y reduce automáticamente la capacidad aprovisionada, pero nunca por debajo de `minRecommendationRequestsPerSecond`. Se produce un breve retraso durante el aumento de la capacidad que podría provocar la pérdida de solicitudes.

Tu factura es la cantidad mínima de solicitudes por hora (en función de minRecommendationRequestsPerSecond) o la cantidad real de solicitudes, que sea mayor. El rendimiento real de las solicitudes utilizado se calcula como el promedio requests/second en un período de una hora. Te recomendamos empezar por el predeterminado`minRecommendationRequestsPerSecond`, realizar un seguimiento de tu uso mediante CloudWatch las métricas de Amazon y, a continuación, aumentarlo `minRecommendationRequestsPerSecond` según sea necesario.

Creación de recomendadores de dominios en Amazon Personalize

Puedes crear recomendaciones con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS SDKs. A continuación, se incluyen pasos detallados para crear recomendadores

con la consola de Amazon Personalize y ejemplos de código que muestran cómo crear un recomendador solo con los campos obligatorios.

- Para ver muestras de código que presentan cómo habilitar los metadatos en las recomendaciones, consulte Habilitación de los metadatos en las recomendaciones.
- Para ver muestras de código que presentan cómo configurar las columnas utilizadas en el entrenamiento de los modelos que respaldan el recomendador, consulte Configuración de las columnas utilizadas al crear un recomendador de dominios de Amazon Personalize.
- Para ver muestras de código que presentan cómo configurar la exploración para los casos de uso de `Top picks for you` y `Recommended for you`, consulte Configuración de la exploración de un recomendador de dominios.

Temas

- Creación de generadores de recomendaciones (consola)
- Creación de un recomendador (AWS CLI)
- Creación de un recomendador (AWS SDKs)

Creación de generadores de recomendaciones (consola)

> **Important**
>
> Un valor elevado de `minRecommendationRequestsPerSecond` aumentará su factura. Le recomendamos empezar con 1 para `minRecommendationRequestsPerSecond` (el valor predeterminado). Realiza un seguimiento de tu consumo mediante CloudWatch las métricas de Amazon y auméntalo `minRecommendationRequestsPerSecond` según sea necesario. Para obtener más información, consulte Solicitudes de recomendación mínimas por segundo y escalado automático.

Cree generadores de recomendaciones para cada uno de sus casos de uso con la consola de Amazon Personalize de la siguiente manera. Si acaba de crear su grupo de conjuntos de datos de dominios y ya se encuentra en la página Información general, vaya al paso 3.

Para crear generadores de recomendaciones

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos de dominio.
3. En el paso 3, elija Usar recomendadores de <nombre de dominio> y, después, Crear recomendadores.
4. En la página Elegir casos de uso, elija los casos de uso en los que desea crear generadores de recomendaciones y asigne a cada uno un Nombre de generador de recomendaciones. Amazon Personalize crea un generador de recomendaciones para cada caso de uso que elija. Los casos de uso disponibles dependen de su dominio. Para obtener información acerca de la elección de un caso de uso, consulte Elección de un caso de uso.
5. Elija Siguiente.
6. En la página Configuración avanzada, configure cada uno de los generadores de recomendaciones en función de las necesidades de su empresa:

   - Para cada conjunto de datos utilizado por el caso de uso del generador de recomendaciones, puede elegir las columnas que Amazon Personalize tiene en cuenta a la hora de entrenar los modelos que respaldan su generador de recomendaciones. De forma predeterminada, Amazon Personalize usa todas las columnas que se pueden usar durante el entrenamiento. Para obtener más información, consulte Configuración de las columnas utilizadas al crear un recomendador de dominios de Amazon Personalize.
   - Puede modificar Solicitudes de recomendación mínimas por segundo para especificar una nueva capacidad mínima de solicitudes para su generador de recomendaciones. Un valor elevado de `minRecommendationRequestsPerSecond` aumentará su factura. Le recomendamos empezar con 1 (el valor predeterminado). Realiza un seguimiento de tu consumo mediante CloudWatch las métricas de Amazon y auméntalo `minRecommendationRequestsPerSecond` según sea necesario. Para obtener más información, consulte Solicitudes de recomendación mínimas por segundo y escalado automático.
   - Si quiere poder incluir metadatos de los conjuntos de datos de elementos en las recomendaciones, seleccione Devolver metadatos de los elementos en los resultados de las recomendaciones. Si está habilitada, puede especificar las columnas del conjunto de datos de elementos en su solicitud de recomendaciones o clasificación personalizada. Amazon Personalize devuelve estos datos para cada elemento en la respuesta a la recomendación.

Para habilitar los metadatos, debe tener un conjunto de datos de elementos con una columna de metadatos.

- Para nuestros casos de `Top picks for your` o `Recommended for you`, si lo desea, puede realizar cambios en la configuración de exploración. La exploración implica probar diferentes recomendaciones de elementos para saber cómo responden los usuarios a elementos con muy pocos datos de interacción. Utilice los siguientes campos para configurar la exploración:
  - Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración, las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).
  - Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

    Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.
- Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.

7. Si quiere crear recomendadores para cada uno de sus casos de uso, elija Crear recomendadores.

Puede supervisar el estado de cada uno de los generadores de recomendaciones en la página Generadores de recomendaciones. Cuando el estado de su generador de recomendaciones sea Activo, podrás usarlo en tu aplicación para obtener recomendaciones.

Creación de un recomendador (AWS CLI)

Usa el siguiente AWS CLI código para crear un recomendante para un caso de uso de dominio. Ejecute este código para cada uno de sus casos de uso de dominio. Para `recipeArn`, proporcione el nombre de recurso de Amazon (ARN) para su caso de uso. Los casos de uso disponibles dependen de su dominio. Para obtener una lista de casos de uso y sus correspondientes, ARNs consulteElección de un caso de uso.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN
```

Creación de un recomendador (AWS SDKs)

Cree un generador de recomendaciones para un caso de uso de dominio con el siguiente código: Asigne un nombre al generador de recomendaciones y proporcione el nombre de recurso de Amazon (ARN) de su grupo de conjuntos de datos de dominio. Para `recipeArn`, proporcione el ARN para su caso de uso. Ejecute este código para cada uno de sus casos de uso de dominio. Los casos de uso disponibles dependen de su dominio. Para obtener una lista de los casos de uso ARNs, sus requisitos y sus requisitos, consulteElección de un caso de uso.

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();
```

Habilitación de los metadatos en las recomendaciones para un recomendador de dominios en Amazon Personalize

> **Important**
>
> Al habilitar los metadatos en las recomendaciones, se incurre en costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Al crear un recomendador, puede habilitar la opción para incluir los metadatos de los elementos de su conjunto de datos de elementos en los resultados de las recomendaciones. Si está habilitada, puede especificar las columnas del conjunto de datos de elementos en su solicitud de recomendaciones. Amazon Personalize devuelve estos datos para cada elemento en la respuesta a la recomendación.

Puede usar metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar los géneros de las películas a los carruseles. También puede usarlos para evaluar la calidad de las recomendaciones de manera visual. Si usa IA generativa en la aplicación, puede conectar los metadatos a las indicaciones de la IA a fin de generar contenido más relevante. Para obtener más información sobre el uso de Amazon Personalize con la IA generativa, consulte Amazon Personalize y la IA generativa.

Para agregar metadatos a las recomendaciones, debe tener un conjunto de datos de elementos con una columna de metadatos. No es necesario usar los metadatos en el entrenamiento. Para obtener información sobre cómo crear un conjunto de datos, consulte Creación de un esquema y un conjunto de datos. Para obtener información sobre la actualización de los datos de un conjunto de datos, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

En los siguientes ejemplos de código se muestra cómo habilitar la opción de incluir metadatos de elementos con AWS CLI o con AWS SDKs. Para hacerlo con la consola de Amazon Personalize, habilite los metadatos en la página Configuración avanzada al crear el recomendador. Para obtener más información, consulte Creación de generadores de recomendaciones (consola).

Habilitación de metadatos (AWS CLI)

Si tiene un conjunto de datos de elementos y quiere tener la opción de incluir metadatos al recibir recomendaciones, establezca `enableMetadataWithRecommendations` en `true` en `recommender-config`.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group \
--recipe-arn recipe ARN \
--recommender-config "{\"enableMetadataWithRecommendations\": "true"}"
```

Habilitar los metadatos (AWS SDKS)

Si tiene un conjunto de datos de elementos y quiere tener la opción de incluir metadatos al recibir recomendaciones, establezca `enableMetadataWithRecommendations` en `true` en `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)
```

Configuración de las columnas utilizadas al crear un recomendador de dominios de Amazon Personalize

Al crear un recomendador, puede modificar las columnas que Amazon Personalize tiene en cuenta al entrenar los modelos que respaldan el recomendador.

Puede hacerlo para experimentar con diferentes combinaciones de datos de entrenamiento. O puede excluir columnas sin datos significativos. Por ejemplo, puede tener una columna que desee utilizar únicamente para filtrar recomendaciones. Puede excluir esta columna del entrenamiento y Amazon Personalize solo la tendrá en cuenta al filtrar.

No puede excluir las columnas EVENT_TYPE. De forma predeterminada, Amazon Personalize usa todas las columnas que se pueden usar durante el entrenamiento. Los siguientes datos siempre se excluyen del entrenamiento:

- Columnas con el tipo de datos booleano
- Datos de impresiones
- Campos de cadena personalizados que no son categóricos ni textuales

No puede incluir datos de impresiones en el entrenamiento, pero si el caso de uso o receta los utiliza, Amazon Personalize usa los datos de impresiones para guiar la exploración cuando reciba recomendaciones.

En los siguientes ejemplos de código se muestra cómo configurar las columnas que se utilizan cuando se entrena con AWS CLI o con. AWS SDKs Para hacerlo con la consola de Amazon Personalize, especifique las columnas que se usarán en la página Configuración avanzada al crear el recomendador. Para obtener más información, consulte Creación de generadores de recomendaciones (consola).

Configuración de las columnas que se utilizan en el entrenamiento (AWS CLI)

Para excluir columnas del entrenamiento, proporcione el objeto `excludedDatasetColumns` en la `trainingDataConfig` como parte de la configuración del generador de recomendaciones. Para cada clave del objeto, proporcione el tipo de conjunto de datos. Para cada valor, proporcione la lista de columnas que se excluirán. Para obtener más información, consulte Configuración de las columnas utilizadas al crear un recomendador de dominios de 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\"]}}}"
```

Configuración de las columnas que se utilizan en el entrenamiento (AWS SDKs)

Para excluir las columnas del entrenamiento, proporcione el objeto `excludedDatasetColumns` en la `trainingDataConfig` como parte de la configuración del generador de recomendaciones. Para cada clave, proporciona el tipo de conjunto de datos. Para cada valor, proporcione la lista de columnas que se excluirán. En el código siguiente se muestra cómo excluir columnas de la formación al crear un programa de recomendaciones. Para obtener más información, consulte Configuración de las columnas utilizadas al crear un recomendador de dominios de 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();
```

Configuración de la exploración de un recomendador de dominios

Para los casos de uso `Top picks for your` o `Recommended for you`, Amazon Personalize utiliza la exploración al recomendar elementos. La exploración implica probar diferentes recomendaciones de elementos para saber cómo responden los usuarios a elementos con muy pocos datos de interacción. Puede configurar la exploración con lo siguiente:

- Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración, las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).

- Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

  Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.

En los siguientes ejemplos de código se muestra cómo configurar la exploración de un recomendante con AWS CLI o. AWS SDKs Para hacerlo con la consola de Amazon Personalize, especifique las columnas de exploración en la página Configuración avanzada al crear el recomendador. Para obtener más información, consulte Creación de generadores de recomendaciones (consola).

Configuración de la exploración (AWS CLI)

En el siguiente código se muestra cómo configurar la exploración al crear un generador de recomendaciones para el caso de uso de `Top picks for you`. El ejemplo utiliza los valores predeterminados.

Si tiene un conjunto de datos de elementos y quiere tener la opción de incluir metadatos al recibir recomendaciones, actualice `recommender-config` para agregar un campo `enableMetadataWithRecommendations` y establézcalo en `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\"}}"
```

Configuración de la exploración (AWS SDKs)

Para los casos de uso `Top picks for you` o `Recommended for you`, Amazon Personalize utiliza la exploración al recomendar elementos. La exploración implica probar diferentes recomendaciones de elementos para saber cómo responden los usuarios a elementos con muy pocos datos de interacción. Puede configurar la exploración con lo siguiente:

- Énfasis en la exploración de los elementos menos relevantes (ponderación de la exploración): configure cuánto explorar. Especifique un valor decimal entre 0 y 1. El valor predeterminado es 0,3. Cuanto más se acerque el valor a 1, mayor será la exploración. Con una mayor exploración, las recomendaciones incluyen más elementos con menos datos de interacciones de elementos o relevancia en función del comportamiento anterior. En cero, no se realiza ninguna exploración y las recomendaciones se basan en los datos actuales (relevancia).
- Límite de antigüedad del elemento de exploración: especifique la antigüedad máxima del elemento en días desde la última interacción entre todos los elementos del conjunto de datos de interacciones de elementos. Esto define el alcance de la exploración del elemento en función de su antigüedad. Amazon Personalize determina la antigüedad del elemento en función de su marca de tiempo de creación o, si faltan los datos de esa marca, de los datos de interacciones de elementos. Para obtener más información sobre cómo Amazon Personalize determina la antigüedad del elemento, consulte Datos de la marca de tiempo de creación.

  Para aumentar el número de elementos que Amazon Personalize considera durante la exploración, indique un valor superior. El mínimo es de 1 día y el valor predeterminado es de 30 días. Las recomendaciones pueden incluir elementos con una antigüedad superior al límite de antigüedad que especifique. Esto se debe a que estos elementos son relevantes para el usuario y la exploración no los identificó.

En el código siguiente se muestra cómo configurar la exploración al crear un generador de recomendación. El ejemplo utiliza los valores predeterminados.

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();
```

Evaluación de un recomendador de dominios de Amazon Personalize

Puede evaluar el rendimiento de su generador de recomendaciones mediante métricas en línea y sin conexión. Las métricas en línea son los resultados empíricos que observa en las interacciones de sus usuarios con las recomendaciones en tiempo real. Por ejemplo, puede registrar la tasa de clics de sus usuarios mientras navegan por su catálogo. Usted es responsable de generar y registrar cualquier métrica en línea.

Las métricas sin conexión son las que Amazon Personalize genera se crea un recomendador. Con las métricas sin conexión, puede evaluar el rendimiento de los modelos que respaldan el recomendador. Puede ver los efectos de modificar la configuración de un recomendador y comparar los resultados de los recomendadores entrenados con diferentes casos de uso con los mismos datos en el mismo grupo de conjuntos de datos.

Evite comparar las métricas de distintos recomendadores entrenados con datos diferentes. La diferencia en las métricas puede deberse a la diferencia en los datos y no al rendimiento del modelo. Por ejemplo, puede tener un grupo de conjuntos de datos con datos de evento `purchase` dispersos para cada usuario y otro con datos de eventos `view` sólidos. En función de métricas como `precision at K`, el recomendador entrenado a partir de los datos de evento de visualización podría parecer que tiene mejor rendimiento debido al mayor número de interacciones.

Para obtener las métricas de rendimiento, Amazon Personalize divide los datos de las interacciones de entrada en un conjunto de entrenamiento y un conjunto de pruebas. El conjunto de entrenamiento consta del 90 % de los datos de sus usuarios y sus interacciones. El conjunto de pruebas está formado por el 10 % restante de los usuarios y sus datos de interacciones.

A continuación, Amazon Personalize crea el generador de recomendaciones con el conjunto de entrenamiento. Una vez finalizado el entrenamiento, Amazon Personalize proporciona al nuevo generador de recomendaciones el 90 % más antiguo de los datos de cada usuario del conjunto de pruebas como entrada. A continuación, Amazon Personalize calcula las métricas comparando las recomendaciones que genera el generador de recomendaciones con las interacciones reales del 10 % más reciente de los datos de cada usuario del conjunto de pruebas.

Temas

- Recuperación de métricas
- Definiciones de métricas
- Ejemplo
- Recursos adicionales

Recuperación de métricas

Una vez que su generador de recomendaciones esté activo, podrá ver las métricas del generador de recomendaciones en la consola de Amazon Personalize o recuperar las métricas llamando a la operación DescribeRecommender.

Temas

- Visualización de métricas (consola)
- Recuperando métricas (AWS CLI)
- Recuperando métricas (AWS SDK)

Visualización de métricas (consola)

Para ver las métricas de los generadores de recomendaciones en la consola, vaya a la página de detalles de su generador de recomendaciones.

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos de dominio.
3. En el panel de navegación, elija Recomendadores.
4. En la lista de generadores de recomendaciones elija uno para ver sus métricas.

Recuperando métricas (AWS CLI)

En el siguiente código se muestra cómo obtener las métricas para un generador de recomendaciones con la CLI AWS CLI.

```
aws personalize describe-recommender \
--recommender-arn recommender arn
```

A continuación se muestra un ejemplo del resultado de las métricas de un generador de recomendaciones creado para el caso práctico Mejores opciones del dominio de 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",
    }
}
```

Recuperando métricas (AWS SDK)

En el siguiente código se muestra cómo obtener las métricas para un generador de recomendaciones con la CLI AWS SDK para Python (Boto3).

```
import boto3

personalize = boto3.client('personalize')

response = personalize.describe_recommender(
    recommenderArn = 'recommender_arn'
)
print(response['recommender']['modelMetrics'])
```

A continuación se muestra un ejemplo del resultado de las métricas de un generador de recomendaciones creado para el caso práctico Mejores opciones del dominio de 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",
    }
}
```


Definiciones de métricas

Las métricas que Amazon Personalize genera para los generadores de recomendaciones se describen a continuación con los siguientes términos:

- La recomendación relevante es una recomendación para un elemento con el que el usuario ha interactuado realmente. Estos elementos provienen del 10 % más reciente de los datos de interacciones de cada usuario del conjunto de pruebas.
- Laclasificación hace referencia a la posición de un elemento recomendado en la lista de recomendaciones. Se supone que la posición 1 (al principio de la lista) es la más importante para el usuario.

Para cada métrica, los números más altos (cercanos a 1) son mejores. Para profundizar aún más, consulte los recursos que se enumeran en Recursos adicionales.

coverage

El valor de la cobertura indica la proporción de elementos únicos que Amazon Personalize podría recomendar del número total de elementos únicos en los conjuntos de datos de interacciones y elementos. Una puntuación de cobertura más elevada significa que Amazon Personalize recomienda más elementos tuyos, en lugar de los mismos elementos una y otra vez para distintos usuarios. Los casos de uso en los que se incluye la exploración de elementos, como Mejores opciones (VIDEO_ON_DEMAND) y Recomendaciones (ECOMMERCE), tienen una mayor cobertura que los que no lo hacen.

rango recíproco medio en 25

Esta métrica indica la capacidad de un modelo para generar una recomendación relevante en la posición mejor clasificada. Puede elegir un modelo con un rango recíproco medio alto de 25 si está generando resultados de búsqueda relevantes para un usuario y no espera que el usuario elija un elemento que se encuentre más abajo en la lista. Por ejemplo, los usuarios suelen elegir la primera receta de cocina en los resultados de búsqueda.

Amazon Personalize calcula esta métrica utilizando la puntuación de rango recíproco medio de las solicitudes de recomendaciones. Cada puntuación de rango recíproco se calcula de la siguiente manera: `1 / the rank of the highest item interacted with by the user`, donde el total de clasificaciones posibles es 25. Se ignoran otros elementos de menor rango con los que interactúa el usuario. Si el usuario eligió el primer elemento, la puntuación es 1. Si no elige ningún elemento, la puntuación es 0.

Por ejemplo, puede mostrar a tres usuarios diferentes 25 recomendaciones para cada uno:

- Si el usuario 1 hace clic en el elemento del rango 4 y en el elemento del rango 10, su puntuación de clasificación recíproca es. 1/4
- Si el usuario 2 hace clic en un elemento del rango 2, en un artículo en el rango 4 y en un elemento del rango 12, su puntuación de clasificación recíproca es. 1/2
- Si el usuario 3 hace clic en un solo elemento del rango 6, su puntuación de clasificación recíproca es. 1/6

El rango recíproco medio de todas las solicitudes de recomendaciones (en este caso, 3) se calcula de la siguiente manera: `(1/4 + 1/2 + 1/6) / 3 = .3056`.

ganancia acumulada descontada normalizada (NDCG) en K (5, 10 o 25)

Esta métrica le indica lo bien que su modelo clasifica las recomendaciones, donde K es un tamaño de muestra de 5, 10 o 25 recomendaciones. Esta métrica es útil si lo que más le interesa es clasificar las recomendaciones más allá del elemento mejor clasificado (para ello, consulte `mean reciprocal rank at 25`). Por ejemplo, la puntuación de `NDCG at 10` sería útil si tiene una aplicación que muestra hasta 10 películas en un carrusel a la vez.

Amazon Personalize calcula la NDCG asignando ponderación a las recomendaciones en función de su posición en la clasificación de cada usuario del conjunto de pruebas. Se descuenta cada recomendación (se le asigna una ponderación inferior) por un factor según su posición. La métrica final es la media de todos los usuarios en el conjunto de pruebas. La ganancia acumulada descontada normalizada en K supone que las recomendaciones que se encuentran más abajo en una lista son menos relevantes que las recomendaciones que están más arriba en la lista.

Amazon Personalize utiliza un factor de ponderación de `1/log(1 + position)`, donde el principio de la lista es la posición `1`.

precisión en K

Esta métrica indica la relevancia de las recomendaciones de su modelo, en función de un tamaño de muestra de K (de 5, 10 o 25) recomendaciones.

Amazon Personalize calcula esta métrica en función del número de recomendaciones relevantes de las K recomendaciones principales para cada usuario de las pruebas, dividido por K, donde K es 5, 10 o 25. La métrica final es la media de todos los usuarios en el conjunto de pruebas.

Por ejemplo, si recomiendas 10 elementos a un usuario y el usuario interactúa con 3 de ellos, la precisión en K es de 3 elementos predichos correctamente divididos por el total de 10 elementos recomendados: `3 / 10 = .30`.

Esta métrica recompensa la recomendación precisa de los elementos relevantes. Cuanto más cerca esté la puntuación a uno, más preciso será el modelo.

Ejemplo

El siguiente es un ejemplo sencillo de un generador de recomendaciones que produce una lista de las recomendaciones para un usuario específico. La segunda y la quinta recomendación coinciden con los registros de los datos de prueba de este usuario. Estas son las recomendaciones pertinentes. Si K se establece en `5`, se generan las siguientes métricas para el usuario.

reciprocal_rank

Cálculo: 1/2

Resultado: 0,5000

normalized_discounted_cumulative_gain_at_5

Cálculo: 1/log ((1 + 2) + 1/log (1 + 5))/1/log((1 + 1) + 1/log (1 + 2))

Resultado: 0,6241

precision_at_5

Cálculo: 2/5

Resultado: 0,4000

Recursos adicionales

Para profundizar en los diferentes tipos de métricas para los sistemas de recomendaciones, consulte los siguientes recursos externos:

- MRR vs MAP vs NDCG: métricas de Rank-Aware evaluación y cuándo usarlas
- Ganancia acumulada descontada: las métricas de clasificación que debe conocer
- Recuperación y precisión en k para sistemas de generadores de recomendaciones
- Clasificación de métricas de evaluación para sistemas de generadores de recomendaciones

Actualización de un recomendador

Tras crear un recomendador, puede actualizar su configuración:

- Puede actualizar las columnas que el recomendador utiliza en el entrenamiento. Si modifica las columnas utilizadas en el entrenamiento, Amazon Personalize inicia automáticamente un reentrenamiento completo de los modelos que respaldan al recomendador. Mientras se completa la actualización, podrá seguir recibiendo recomendaciones del recomendador. El recomendador utiliza la configuración anterior hasta que se complete la actualización. Para realizar un seguimiento del estado de esta actualización, utilice el `latestRecommenderUpdate` devuelto en la operación DescribeRecommender. Si proporciona las mismas columnas que proporcionó cuando creó el recomendador, no se realizará ninguna actualización.

- Puede actualizar las solicitudes de recomendación mínimas del recomendador por segundo. Esto especifica el rendimiento básico de las solicitudes de recomendación que proporciona Amazon Personalize. Un valor elevado aumentará su factura. Le recomendamos empezar con 1. Realice un seguimiento de su uso mediante las métricas de Amazon CloudWatch y aumente el valor según sea necesario. Para obtener más información, consulte Solicitudes de recomendación mínimas por segundo y escalado automático.
- Para los casos de uso Mejores opciones y Recomendaciones, puede actualizar la configuración de exploración ajustando el énfasis en la exploración de los elementos relevantes y el límite de antigüedad de los elementos de exploración. Para obtener información sobre la exploración, consulta la sección correspondiente a su caso de uso en Elección de un caso de uso.

Puede actualizar los recomendadores con la consola de Amazon Personalize, AWS Command Line Interface (AWS CLI) o con los SDK de AWS.

Temas

- Actualización de un recomendador (consola de Amazon Personalize)
- Actualización de un recomendador (AWS CLI)
- Actualización de un recomendador (SDK de AWS)

Actualización de un recomendador (consola de Amazon Personalize)

Después de crear un recomendador, lo puede actualizar. Puede actualizar las columnas que el recomendador utiliza en el entrenamiento y las solicitudes de recomendación mínimas por segundo del recomendador. Para los casos de uso Mejores opciones y Recomendaciones, actualizar la configuración de exploración. Para actualizar un recomendador con la consola, haga lo siguiente.

Para actualizar la configuración de un recomendador (consola)

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos de dominio.
3. En el panel de navegación, elija Recomendadores.
4. En la página Recomendadores, elija el recomendador que desea actualizar.
5. En la configuración del recomendador, seleccione Editar.

6. Cambie la configuración del recomendador y seleccione Actualizar. Para obtener información acerca de las distintas opciones de configuración, consulte Creación de generadores de recomendaciones (consola).

Actualización de un recomendador (AWS CLI)

Para actualizar el recomendador con AWS CLI, utilice el comando `update-recommender`. Proporcione el nombre de recurso de Amazon (ARN) para el recomendador y la configuración actualizada. El siguiente código muestra cómo actualizar las columnas que un recomendador utiliza para el entrenamiento.

```
aws personalize update-recommender \
--dataset-group-arn dataset group ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

Si modifica las columnas utilizadas en el entrenamiento, Amazon Personalize inicia automáticamente un reentrenamiento completo de los modelos que respaldan al recomendador. Mientras se completa la actualización, podrá seguir recibiendo recomendaciones del recomendador. El recomendador utiliza la configuración anterior hasta que se complete la actualización. Para realizar un seguimiento del estado de esta actualización, utilice el `latestRecommenderUpdate` devuelto en la operación DescribeRecommender.

Para obtener más información acerca de las distintas opciones que puede cambiar, consulte RecommenderConfig.

Actualización de un recomendador (SDK de AWS)

Para actualizar el recomendador con AWS, utilice la operación UpdateRecommender. Proporcione el nombre de recurso de Amazon (ARN) para el recomendador y especifique la nueva configuración. El siguiente código muestra cómo actualizar las columnas que un recomendador utiliza para el entrenamiento.

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 modifica las columnas utilizadas en el entrenamiento en `excludedDatasetColumns` de `recommenderConfig`, Amazon Personalize inicia automáticamente un reentrenamiento completo de los modelos que respaldan al recomendador. Mientras se completa la actualización, podrá seguir recibiendo recomendaciones del recomendador. El recomendador utiliza la configuración anterior hasta que se complete la actualización. Para realizar un seguimiento del estado de esta actualización, utilice el `latestRecommenderUpdate` devuelto en la operación DescribeRecommender.

Para obtener más información acerca de las distintas opciones que puede cambiar, consulte RecommenderConfig.

Detención de un recomendador

Una vez que el recomendador esté activo, puede detenerlo e iniciarlo más tarde. De esta forma, puede pausar la facturación del recomendador y pagarlo solo cuando lo utilice. Por ejemplo, es posible que deba obtener recomendaciones solo en ciertos días de la semana. Puede detener el recomendador los días que no lo necesite y, después, iniciarlo los días que sí lo necesite.

Una vez que detenga un recomendador, no podrá usarlo para obtener recomendaciones. La detención de un recomendador detiene la facturación y el reentrenamiento. Sin embargo, detener un recomendador no lo elimina. Puede reiniciarlo en cualquier momento y volver a recibir recomendaciones. Al iniciar un recomendador, no se crea uno nuevo con los datos. Más bien, reanuda la facturación del recomendador y el reentrenamiento cada 7 días.

Puede detener e iniciar un recomendador con la consola de Amazon Personalize, AWS Command Line Interface (AWS CLI) y los SDK de AWS.

Estados del recomendador

Al detener un recomendador, su estado cambia de ACTIVE a INACTIVE en la siguiente secuencia:

ACTIVE > STOP PENDING > STOP IN PROGRESS > INACTIVE

Al iniciar un recomendador, su estado cambia de ACTIVE a INACTIVE en la siguiente secuencia:

INACTIVE > START PENDING > START IN PROGRESS > ACTIVE

Temas

- Detención e inicio de un recomendador (consola)
- Detención de un recomendador (AWS CLI)
- Detención de un recomendador (SDK de AWS)

Detención e inicio de un recomendador (consola)

Puede detener un recomendador desde la página de detalles del recomendador en la consola de Amazon Personalize.

Para detener un recomendador

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos de dominio.
3. En el panel de navegación, elija Recomendadores.
4. En la página Recomendadores, elija el recomendador que desea detener.
5. En la página de detalles del recomendador, elija Detener el recomendador en la parte superior derecha y confirme en la ventana que aparece. Si el estado del recomendador está inactivo, significa que el recomendador se ha detenido. Puede volver a iniciarlo desde la misma página.

Detención de un recomendador (AWS CLI)

Para detener un recomendador activo con la AWS CLI, utilice el comando `stop-recommender`, que usa la operación de la API StopRecommender, y proporcione el nombre de recurso de Amazon (ARN) para el recomendador. Para reiniciarlo, puede utilizar el comando `start-recommender`, que usa StartRecommender. El código siguiente muestra cómo detener un recomendador:

```
aws personalize stop-recommender --recommender-arn "recommender arn"
```

Detención de un recomendador (SDK de AWS)

Para detener un recomendador activo con los SDK de AWS, utilice la operación de la API StopRecommender y proporcione el nombre de recurso de Amazon (ARN) para el recomendador. Para reiniciarlo, utilice StartRecommender. El código siguiente muestra cómo detener un recomendador:

SDK for Python (Boto3)

Para detener un recomendador activo con el SDK para Python (Boto3), utilice el método `stop_recommender` y proporcione el nombre de recurso de Amazon (ARN) para el recomendador de la siguiente manera:

```
import boto3
personalize = boto3.client('personalize')

stop_recommender_response = personalize.stop_recommender(
    recommenderArn = "recommenderARN"
)
print(stop_recommender_response)
```

SDK for Java 2.x

Para detener un recomendador activo con el SDK para Java 2.x, utilice el método `stopRecommender` y proporcione el nombre de recurso de Amazon (ARN) para el recomendador de la siguiente manera:

```
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();
```

Recursos personalizados para el entrenamiento y la implementación de modelos de Amazon Personalize

Si utiliza una receta personalizada, después de importar los datos, podrá crear los recursos personalizados para entrenar e implementar los modelos de Amazon Personalize. Estos recursos se usan para obtener recomendaciones. Al crear recursos personalizados, haga lo siguiente:

1. Cree y configure una solución: personalice los parámetros de la solución y los hiperparámetros específicos de receta para que el modelo satisfaga sus necesidades empresariales específicas. De forma predeterminada, las nuevas versiones de solución utilizan el entrenamiento automático para crear versiones de solución con una frecuencia configurable. La frecuencia predeterminada es cada siete días. Para obtener más información sobre la configuración de una solución, consulte Configuración de una solución. Para obtener más información sobre las recetas personalizadas en Amazon Personalize, consulte Elección de una receta.
2. Crear una versión de solución (para soluciones que no utilizan el entrenamiento automático): en el caso de las soluciones que utilizan el entrenamiento automático, la creación de la versión de solución se inicia automáticamente una vez que la solución está activa. En el caso de las soluciones que utilizan el entrenamiento manual, la versión de solución se crea manualmente. La versión de la solución genera recomendaciones o segmentos de usuarios de Amazon Personalize. Para obtener más información sobre la creación manual de una versión de solución, consulte Creación manual de una versión de solución. Para detener la creación de la versión de la solución, vaya a la página de detalles de la versión de la solución y elija Detener. Para obtener más información, consulte Detención de la creación de una versión de solución.
3. Evalúe la versión de la solución: utilice las métricas que Amazon Personalize genera a partir de la nueva versión de la solución para evaluar el rendimiento del modelo. Consulte Evaluación de una versión de solución de Amazon Personalize con métricas.
4. Implemente la versión de la solución con una campaña (solo para las recomendaciones en tiempo real): cree una campaña para implementar la versión de la solución. La campaña se utiliza cuando solicita generadores de recomendaciones en tiempo real. Si recibe recomendaciones por lotes, no es necesario crear una campaña. Para obtener más información, consulte Implementación de una versión de solución de Amazon Personalize con una campaña. Si desea cambiar la configuración de una campaña existente, por ejemplo, habilitar los metadatos en las recomendaciones, debe actualizar la campaña. Para obtener más información, consulte Actualización de la configuración de una campaña de Amazon Personalize.

Temas

- Configuración de una solución personalizada en Amazon Personalize
- Actualización de una solución para cambiar la configuración de entrenamiento automático
- Creación manual de una versión de solución
- Detención de la creación de una versión de solución
- Evaluación de una versión de solución de Amazon Personalize con métricas
- Implementación de una versión de solución de Amazon Personalize con una campaña
- Actualización de la configuración de una campaña de Amazon Personalize

Configuración de una solución personalizada en Amazon Personalize

Una vez que termine de importar datos, estará listo para crear una solución. Una solución hace referencia a la combinación de una receta de Amazon Personalize, parámetros de entrenamiento personalizados y una o más versiones de solución. Una versión de la solución hace referencia a un modelo de machine learning entrenado.

De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático para crear una nueva versión de solución cada siete días. El entrenamiento automático solo se produce si ha importado datos de interacción masivos o en tiempo real desde el último entrenamiento. Esto incluye interacciones de elemento o, en el caso de soluciones que utilizan la receta Next-Best-Action, los datos de interacciones de elemento. El entrenamiento automático continúa hasta que elimine la solución. Para obtener más información, consulte Configuración del entrenamiento automático.

Si ya tiene una solución, puede utilizar la consola de Amazon Personalize para clonar la solución. Al clonar una solución, puede utilizar la configuración de la solución existente como punto de partida, como la receta y los hiperparámetros, y realizar cambios. Para obtener más información, consulte Clonación de una solución (consola).

Puede crear y configurar una solución mediante la consola, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Después de crear una solución, puede ver los detalles de configuración en la página de detalles de la solución de la consola de Amazon Personalize, o con la operación DescribeSolution.

De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa.

Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Temas

- Creación de una solución
- Configuración del entrenamiento automático
- Configuración de las columnas que se utilizan en el entrenamiento
- Optimización de una solución para un objetivo adicional
- Optimización de una solución con la configuración de eventos
- Hiperparámetros y HPO
- Elección de los datos de interacciones de elementos utilizados para el entrenamiento
- Clonación de una solución (consola)

Creación de una solución

Puede crear una solución personalizada con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. A continuación, se incluyen pasos detallados para crear una solución con la consola de Amazon Personalize y ejemplos de código que muestran cómo crear una solución solo con los campos obligatorios.

Temas

- Creación de una solución (consola)
- Creación de una solución (AWS CLI)
- Creación de una solución (SDK de AWS)

Creación de una solución (consola)

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la

solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Para crear una solución en la consola, elija el grupo de conjuntos de datos y, a continuación, especifique el nombre de la solución, la receta y la configuración de entrenamiento opcional.

Para configurar una solución (consola)

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en la cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En la página Descripción general, para Paso 3, realice una de las siguientes acciones:
   - Si ha creado un grupo de conjuntos de datos de dominio, elija Usar recursos personalizados y, después, Crear soluciones.
   - Si ha creado un grupo de conjuntos de datos personalizados, elija Crear soluciones.
4. En Nombre de la solución, especifique un nombre para la solución.
5. En Tipo de solución, elija el tipo de solución que desea crear. El tipo que elija determina las recetas que están disponibles.
   - Elija Recomendación de elementos para obtener recomendaciones de elementos para sus usuarios. Por ejemplo, recomendaciones personalizadas sobre películas.
   - Elija Recomendación de acción a fin obtener recomendaciones sobre acciones para los usuarios. Por ejemplo, genere la siguiente mejor acción para un usuario, como descargar la aplicación.
   - Elija Segmentación de usuarios para obtener segmentos de usuarios (grupos de usuarios) en función de los datos de los elementos.
6. En Receta, elija una receta (consulte Elección de una receta).
7. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
8. Elija Siguiente.
9. En la página Configuración de entrenamiento, personalice la solución para que se adapte a los requisitos empresariales.

- En Entrenamiento automático, elija si la solución utiliza el entrenamiento automático. Si usa entrenamiento automático, puede cambiar la `Automatic training frequency`. La frecuencia de entrenamiento predeterminada es cada siete días.

  Recomendamos utilizar el entrenamiento automático. Esto le facilita mantener la relevancia de la recomendación. La frecuencia de entrenamiento depende de los requisitos empresariales, de la receta que utilice y de la frecuencia con que importe los datos. Para obtener más información, consulte Configuración del entrenamiento automático. Para obtener información sobre cómo mantener la relevancia, consulte Mantenimiento de la relevancia de las recomendaciones.
- En Configuración de hiperparámetros, configure cualquier opción de hiperparámetro en función de la receta y de las necesidades empresariales. Las distintas recetas usan diferentes hiperparámetros. Para los hiperparámetros que tiene disponibles, consulte las recetas individuales en Elección de una receta.
- En Columnas para entrenamiento, si la receta genera recomendaciones de elemento o segmentos de usuarios, puede elegir las columnas que Amazon Personalize tiene en cuenta a la hora de entrenar las versiones de solución. Para obtener más información, consulte Configuración de las columnas que se utilizan en el entrenamiento.
- En Configuración adicional, si el conjunto de datos de interacciones con artículos tiene la columna EVENT_TYPE o tanto EVENT_TYPE como EVENT_VALUE, si lo desea, utilice los campos Tipo de evento y Umbral del valor del evento para elegir los datos de interacciones con artículos que Amazon Personalize usará al entrenar el modelo. Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento.

  Si tiene varios tipos de evento y usa la receta User-Personalization-v2 o Personalized-Ranking-v2, también puede especificar ponderaciones distintas para los diferentes tipos. Por ejemplo, puede configurar una solución para dar más peso a los eventos de compra que a los de clic. Para obtener más información, consulte Optimización de una solución con la configuración de eventos.
- Si utiliza la receta Receta User-Personalization o Receta Personalized-Ranking, si lo desea, especifique un Objetivo y elija una Sensibilidad de objetivo con el fin de optimizar su solución en función de un objetivo, además de su relevancia. La sensibilidad de los objetivos configura la forma en que Amazon Personalize equilibra la recomendación de elementos en función del objetivo según la relevancia a través de los datos de interacciones. Para obtener más información, consulte Optimización de una solución para un objetivo adicional.

10. Seleccione Siguiente y revise los detalles de la solución. No puede cambiar la configuración de la solución después de crearla.
11. Elija Create solution (Crear solución). Después de crear una solución, Amazon Personalize comienza a crear la primera versión de solución en una hora. Cuando comience el entrenamiento, podrá supervisarla en la sección Versiones de solución de la página de detalles de la solución. Las versiones de solución creadas automáticamente tienen un Tipo de entrenamiento de AUTOMATIC.

    Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

    - Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
    - Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Creación de una solución (AWS CLI)

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Para crear una solución con la AWS CLI, utilice el comando `create-solution`. Este comando utiliza la operación de la API CreateSolution. En el siguiente código se muestra cómo crear una solución que usa entrenamiento automático. Crea automáticamente una nueva versión de solución cada cinco días.

Para usar el código, actualícelo para asignar un nombre a la solución, especifique el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos, cambie la frecuencia de entrenamiento y especifique el ARN de la receta que se usará. Para obtener información acerca de las recetas, consulte Elección de una receta.

```
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)\"}}"
```

- Recomendamos que utilice el entrenamiento automático. Esto le facilita mantener y mejorar la relevancia de la recomendación. De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. La frecuencia de entrenamiento predeterminada es cada siete días. La frecuencia de entrenamiento depende de los requisitos empresariales, de la receta que utilice y de la frecuencia con que importe los datos. Para obtener más información, consulte Configuración del entrenamiento automático.
- En función de la receta, puede modificar el código para configurar determinadas propiedades e hiperparámetros de la receta (consulte Hiperparámetros y HPO), configurar qué columnas se utilizan para el entrenamiento (consulte Configurar las columnas que se utilizan durante el entrenamiento (AWS CLI)) o filtrar los datos de interacciones de elemento que se utilizan para el entrenamiento (consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento).
- Si utiliza la receta Receta User-Personalization o Receta Personalized-Ranking, puede optimizar la solución para un objetivo, además de por la relevancia. Para obtener más información, consulte Optimización de una solución para un objetivo adicional.

Después de crear la solución, registre el ARN de la solución para usarlo en el futuro. Con el entrenamiento automático, la creación de versiones de la solución comienza una hora después de que el estado de la solución sea ACTIVE. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático. Una vez que comience el entrenamiento, puede obtener el nombre de recurso de Amazon (ARN) de la versión de solución con la operación de API ListSolutionVersions. Para obtener el estado, utilice la operación de la API DescribeSolutionVersion.

Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

- Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
- Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Creación de una solución (SDK de AWS)

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Para crear una solución con los SDK de AWS, utilice la operación de la API CreateSolution. En el siguiente código se muestra cómo crear una solución que usa entrenamiento automático. Crea automáticamente una nueva versión de solución cada cinco días.

Para usar el código, actualícelo para asignar un nombre a la solución, especifique el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos, cambie la frecuencia de entrenamiento y especifique el ARN de la receta que desea usar. Para obtener información acerca de las recetas, consulte Elección de una receta.

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();
```

- Recomendamos que utilice el entrenamiento automático. Esto le facilita mantener y mejorar la relevancia de la recomendación. De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. La frecuencia de entrenamiento predeterminada es cada siete días. La frecuencia de entrenamiento depende de los requisitos empresariales, de la receta que utilice y de la frecuencia con que importe los datos. Para obtener más información, consulte Configuración del entrenamiento automático.
- En función de la receta, puede modificar el código para configurar determinadas propiedades e hiperparámetros de la receta (consulte Hiperparámetros y HPO), configurar qué columnas se utilizan para el entrenamiento (consulte Configuración de las columnas utilizadas durante el entrenamiento (AWS (SDK)) o filtrar los datos de interacciones de elemento que se utilizan para el entrenamiento (consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento).
- Si utiliza la receta Receta User-Personalization o Receta Personalized-Ranking, puede optimizar la solución para un objetivo, además de por la relevancia. Para obtener más información, consulte Optimización de una solución para un objetivo adicional.

Después de crear la solución, registre el ARN de la solución para usarlo en el futuro. Con el entrenamiento automático, la creación de versiones de la solución comienza una hora después de que el estado de la solución sea ACTIVE. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático. Una vez que comience el entrenamiento, puede obtener el nombre de recurso de Amazon (ARN) de la versión de solución con la operación de API ListSolutionVersions. Para obtener el estado, utilice la operación de la API DescribeSolutionVersion.

Puede utilizar el siguiente código Python para esperar a que comience el entrenamiento automático. El método `wait_for_training_to_start` devuelve el ARN de la primera versión de solución.

```
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)
```

Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

- Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
- Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Configuración del entrenamiento automático

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Al crear una solución, puede configurar si la solución utiliza el entrenamiento automático. También puede configurar la frecuencia de entrenamiento. Por ejemplo, puede configurar la solución para crear una nueva versión de solución cada cinco días.

De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático para crear una nueva versión de solución cada siete días. El entrenamiento automático solo se produce si ha importado datos de interacción masivos o en tiempo real desde el último entrenamiento. Esto incluye interacciones de elemento o, en el caso de soluciones que utilizan la receta Next-Best-Action, los datos de interacciones de elemento. El entrenamiento automático continúa hasta que elimine la solución.

Recomendamos que utilice el entrenamiento automático. Facilita el mantenimiento de la solución. Elimina el entrenamiento manual necesario para que la solución aprenda a partir de los datos más recientes. Sin un entrenamiento automático, debe crear de forma manual nuevas versiones de solución para que la solución aprenda a partir de los datos más recientes. Esto puede dar lugar a recomendaciones obsoletas y a una tasa de conversión más baja. Para obtener más información sobre el mantenimiento de las recomendaciones de Amazon Personalize, consulte Mantenimiento de la relevancia de las recomendaciones.

Puede configurar el entrenamiento automático con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Para ver los pasos de la configuración del entrenamiento automático con la consola, consulte Creación de una solución (consola).

Después de crear la solución, registre el ARN de la solución para usarlo en el futuro. Con el entrenamiento automático, la creación de versiones de la solución comienza una hora después de que el estado de la solución sea ACTIVE. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático. Una vez que comience el entrenamiento, puede obtener el nombre de recurso de Amazon (ARN) de la versión de solución

con la operación de API ListSolutionVersions. Para obtener el estado, utilice la operación de la API DescribeSolutionVersion.

Temas

- Directrices y requisitos
- Configuración del entrenamiento automático (AWS CLI)
- Configuración del entrenamiento automático (SDK)

Directrices y requisitos

A continuación, se indican las directrices y los requisitos para el entrenamiento automático:

- El entrenamiento automático solo se produce si ha importado datos de interacción masivos o en tiempo real desde el último entrenamiento. Esto incluye interacciones de elemento o, en el caso de soluciones que utilizan la receta Next-Best-Action, los datos de interacciones de elemento.
- En cada entrenamiento se tienen en cuenta todos los datos del grupo de conjuntos de datos que incluya en el entrenamiento. Para obtener información sobre la configuración de las columnas utilizadas en el entrenamiento, consulte Configuración de las columnas que se utilizan en el entrenamiento.
- Aún puede crear manualmente versiones de solución.
- El entrenamiento automático comienza una hora después de que la solución esté activa. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático.
- La programación del entrenamiento se basa en la fecha de inicio del entrenamiento. Por ejemplo, si la primera versión de solución comienza a entrenar a las 19:00 h y utiliza el entrenamiento semanal, la siguiente versión de solución empezará a entrenarse una semana después, a las 19:00 h.
- Para todas las recetas, recomendamos una frecuencia de entrenamiento semanal como mínimo. Puede especificar una frecuencia de entrenamiento entre 1 y 30 días. El valor predeterminado es cada siete días.
  - Si utiliza User-Personalization-v2, User-Personalization o Next-Best-Action, la solución se actualiza automáticamente para tener en cuenta nuevos elementos o acciones para las recomendaciones. Las actualizaciones automáticas no son lo mismo que el entrenamiento automático. Una actualización automática no crea una versión de solución completamente nueva y el modelo no aprende a partir de los datos más recientes. Para mantener la solución,

la frecuencia de entrenamiento debe seguir siendo de una vez a la semana como mínimo. Para obtener más información sobre las actualizaciones automáticas, incluidas las directrices y los requisitos adicionales, consulte Actualizaciones automáticas.

- Si usa Trending-Now, Amazon Personalize identifica automáticamente los elementos más populares en sus datos de interacciones durante un intervalo de tiempo configurable. Trending-Now puede recomendar elementos agregados desde el último entrenamiento mediante datos de interacciones masivos o en streaming. La frecuencia de entrenamiento debe seguir siendo de una vez a la semana como mínimo. Para obtener más información, consulte Receta Trending-Now.
- Si no utiliza una receta con actualizaciones automáticas ni la receta Trending-Now, Amazon Personalize tendrá en cuenta los elementos nuevos para las recomendaciones solo después del siguiente entrenamiento. Por ejemplo, si utiliza la receta Similar-Items y agrega nuevos elementos cada día, debería utilizar una frecuencia de entrenamiento diaria para que estos elementos aparezcan en las recomendaciones ese mismo día.

Configuración del entrenamiento automático (AWS CLI)

En el siguiente código se muestra cómo crear una solución que cree automáticamente una versión de solución cada cinco días. Para desactivar el entrenamiento automático, establezca `perform-auto-training` en `false`.

Para cambiar la frecuencia de entrenamiento, puede modificar `schedulingExpression` en `autoTrainingConfig`. La expresión debe tener el formato `rate(`*`value unit`*`)`. Para el valor, especifique un número comprendido entre 1 y 30. Para la unidad, especifique `day` o `days`.

Para obtener una explicación completa del comando `create-solution`, consulte Creación de una solución (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)\"}}"
```

Configuración del entrenamiento automático (SDK)

En el siguiente código se muestra cómo crear una solución con entrenamiento automático con los SDK de AWS. La solución crea automáticamente una versión de solución cada cinco días. Para desactivar el entrenamiento automático, establezca `performAutoTraining` en `false`.

Para cambiar la frecuencia de entrenamiento, puede modificar `schedulingExpression` en `autoTrainingConfig`. La expresión debe tener el formato `rate(value unit)`. Para el valor, especifique un número comprendido entre 1 y 30. Para la unidad, especifique `day` o `days`.

Para obtener una explicación completa de la operación de la API CreateSolution, consulte Creación de una solución (SDK de AWS).

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();
```

Puede utilizar el siguiente código Python para esperar a que comience el entrenamiento automático. El método `wait_for_training_to_start` devuelve el ARN de la primera versión de solución.

```
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)
```

Configuración de las columnas que se utilizan en el entrenamiento

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Si su receta genera recomendaciones de elementos o segmentos de usuarios, puede modificar las columnas que Amazon Personalize tiene en cuenta al crear una versión de la solución (entrenamiento de un modelo).

Puede cambiar las columnas que se utilizan durante el entrenamiento para controlar los datos que Amazon Personalize utiliza al entrenar un modelo (creación de una versión de solución). Puede hacerlo para experimentar con diferentes combinaciones de datos de entrenamiento. O puede excluir columnas sin datos significativos. Por ejemplo, puede tener una columna que desee utilizar únicamente para filtrar recomendaciones. Puede excluir esta columna del entrenamiento y Amazon Personalize solo la tendrá en cuenta al filtrar.

No puede excluir las columnas EVENT_TYPE. De forma predeterminada, Amazon Personalize usa todas las columnas que se pueden usar durante el entrenamiento. Los siguientes datos siempre se excluyen del entrenamiento:

- Columnas con el tipo de datos booleano
- Datos de impresiones
- Campos de cadena personalizados que no son categóricos ni textuales

No puede incluir datos de impresiones en el entrenamiento, pero si el caso de uso o receta los utiliza, Amazon Personalize usa los datos de impresiones para guiar la exploración cuando reciba recomendaciones.

Si ya ha creado una solución y desea modificar las columnas que utiliza durante el entrenamiento, puede clonar la solución. Al clonar una solución, puede utilizar la configuración de la solución existente como punto de partida, como la receta y los hiperparámetros, y realizar los cambios necesarios. Para obtener más información, consulte Clonación de una solución (consola).

Puede configurar las columnas que Amazon Personalize utiliza cuando entrena con la consola, AWS Command Line Interface (AWS CLI) o el AWS SDK de Amazon Personalize. Para obtener información acerca de cómo elegir columnas con la consola de Amazon Personalize, consulte los pasos de configuración avanzada en Creación de una solución (consola). Tras crear una solución, puede ver las columnas que utiliza la solución en la página de detalles de la solución de la consola de Amazon Personalize, o con la operación DescribeSolution.

Temas

- Configurar las columnas que se utilizan durante el entrenamiento (AWS CLI)
- Configuración de las columnas utilizadas durante el entrenamiento (AWS (SDK)

Configurar las columnas que se utilizan durante el entrenamiento (AWS CLI)

Para excluir las columnas del entrenamiento, proporcione el objeto `excludedDatasetColumns` en el `trainingDataConfig` como parte de la configuración de la solución. Para cada clave, proporciona el tipo de conjunto de datos. Para cada valor, proporcione la lista de columnas que se excluirán. El código siguiente muestra cómo excluir columnas del entrenamiento al crear una solución con 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\"]}}}"
```

Configuración de las columnas utilizadas durante el entrenamiento (AWS (SDK)

Para excluir las columnas del entrenamiento, proporcione el objeto `excludedDatasetColumns` en el `trainingDataConfig` como parte de la configuración de la solución. Para cada clave, proporciona el tipo de conjunto de datos. Para cada valor, proporcione la lista de columnas que se excluirán. El código siguiente muestra cómo excluir columnas del entrenamiento al crear una solución con AWS SDK para 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)
```

Optimización de una solución para un objetivo adicional

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Si usa la receta User-Personalization o Personalized-Ranking, puede optimizar una solución de Amazon Personalize para alcanzar un objetivo junto con la máxima relevancia, como maximizar los ingresos.

Con las recetas de recomendación de elementos, el objetivo principal de Amazon Personalize es predecir los elementos más relevantes para sus usuarios en función de los datos de interacciones de elementos históricos y en tiempo real. Estos son los elementos con los que es más probable que tus usuarios interactúen (por ejemplo, los artículos en los que probablemente hagan clic). Si tiene un objetivo adicional, como maximizar los minutos de streaming o aumentar los ingresos, puede crear una solución que genere recomendaciones basadas tanto en la relevancia como en su objetivo.

Para optimizar una solución para un objetivo adicional, cree una nueva solución con la receta User-Personalization o la receta Personalized-Ranking y elija la columna de metadatos numéricos del conjunto de datos de elementos que esté relacionada con su objetivo. Al generar recomendaciones, Amazon Personalize da más importancia a los elementos con valores más altos para esta columna de datos. Por ejemplo, puede elegir una columna VIDEO_LENGTH para maximizar los minutos de streaming o una columna PRICE para maximizar los ingresos.

Puedes usar la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS SDKs. Para obtener información sobre el uso de la consola de Amazon Personalize, consulte Creación de una solución (consola).

Temas

- Directrices y requisitos
- Equilibrio del énfasis y la relevancia del objetivo
- Medición del rendimiento de optimización
- Optimización de una solución (AWS CLI)
- Optimización de una solución (AWS SDKs)
- Cuaderno de Jupyter de ejemplo

Directrices y requisitos

Los requisitos del objetivo son los siguientes:

- Solo puede elegir una columna para su objetivo.
- La columna debe tener un tipo numérico en el esquema.
- La columna no puede tener ningún tipo `null` en su esquema.

Para obtener más información sobre esquemas y tipos de datos, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Equilibrio del énfasis y la relevancia del objetivo

Puede haber una compensación al recomendar elementos basándose más en su objetivo que en su relevancia. Por ejemplo, si quiere aumentar los ingresos mediante recomendaciones, las recomendaciones que solo incluyan elementos caros pueden hacer que los artículos sean menos relevantes para sus usuarios y reducir la participación y la conversión de los usuarios.

Para configurar el equilibrio entre la relevancia y su objetivo, elija uno de los siguientes niveles de sensibilidad objetiva al crear la solución:

- Desactivado: Amazon Personalize utiliza principalmente datos de interacciones de elementos para predecir los elementos más relevantes para el usuario.
- Bajo: Amazon Personalize pone menos énfasis en su objetivo. La relevancia a través de los datos de interacciones de elementos es más importante.
- Medio: Amazon Personalize pone el mismo énfasis en su objetivo y relevancia a través de los datos de interacciones de elementos.
- Alto: Amazon Personalize pone más énfasis en su objetivo. La relevancia a través de los datos de interacciones de elementos es menos importante.

Medición del rendimiento de optimización

Cuando crea una versión de solución (entrena un modelo) para una solución con un objetivo de optimización, Amazon Personalize genera una métrica `average_rewards_at_k`. La puntuación de `average_rewards_at_k` indica el rendimiento de la versión de la solución en cuanto al logro de su objetivo. Para calcular esta métrica, Amazon Personalize calcula las recompensas para cada usuario de la siguiente manera:

```
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
```

El `average_rewards_at_k` final es el promedio de todos los valores `rewards_per_user` normalizados para que sea un valor decimal menor o igual a 1 y mayor que 0. Cuanto más se acerque el valor a 1, más beneficios de media por usuario se pueden esperar de las recomendaciones.

Por ejemplo, si su objetivo es maximizar los ingresos de los clics, Amazon Personalize calcula la puntuación de cada usuario dividiendo los ingresos totales generados por los elementos en los que el usuario hizo clic de sus 25 recomendaciones más caras entre los ingresos de todos los elementos recomendados en los que hizo clic el usuario. Amazon Personalize devuelve entonces un promedio normalizado de todas las puntuaciones de los usuarios. Cuanto más cerca esté `average_rewards_at_k` de 1, más ingresos en promedio se pueden esperar por usuario a partir de las recomendaciones.

Para obtener más información sobre la generación de métricas, consulte Evaluación de una versión de solución de Amazon Personalize con métricas.

Optimización de una solución (AWS CLI)

Solo puede optimizar para un objetivo con la receta User-Personalization o Personalized-Ranking. Para optimizar una solución para un objetivo adicional mediante el AWS CLI, cree una nueva solución y especifique los detalles del objetivo con la `optimizationObjective` clave del `solutionConfig` objeto. El valor `optimizationObjective` tiene los siguientes campos:

- `itemAttribute`: especifique el nombre de la columna de metadatos numéricos del conjunto de datos de elementos relacionado con su objetivo.
- `objectiveSensitivity`: especifique el nivel de énfasis que la solución pone en su objetivo al generar recomendaciones. El nivel de sensibilidad de los objetivos configura la forma en que Amazon Personalize equilibra la recomendación de elementos en función de su objetivo frente a la relevancia a través de los datos de interacciones de elementos. El valor `objectiveSensitivity` puede ser`OFF`, LOW, `MEDIUM` o `HIGH`. Para obtener más información, consulte Equilibrio del énfasis y la relevancia del objetivo.

A continuación, se muestra un ejemplo del comando `create-solution` AWS CLI . Reemplace `solution name`, `dataset group arn` y `recipe arn` por los suyos.

Para `optimizationObjective`, sustituya `COLUMN_NAME` por el nombre de la columna numérica de metadatos del conjunto de datos de elementos que esté relacionada con su objetivo. Para`objectiveSensitivity`, especifique OFF, LOW, MEDIUM o 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\"}}"
```

Cuando la solución esté lista, cree una nueva versión de la solución (para ver un ejemplo de comando, consulte Creación de una solución (AWS CLI)). Una vez que haya creado una versión de la solución, podrá ver el rendimiento de la optimización con las métricas de la versión de la solución. Consulte Medición del rendimiento de optimización.

Optimización de una solución (AWS SDKs)

Solo puede optimizar para un objetivo con la receta User-Personalization o Personalized-Ranking.

Para optimizar una solución para un objetivo adicional mediante el AWS SDKs, cree una nueva solución y especifique los detalles del objetivo con la `optimizationObjective` clave del `solutionConfig` objeto de la solución. El valor `optimizationObjective` tiene los siguientes campos:

- `itemAttribute`: especifique el nombre de la columna de metadatos numéricos del conjunto de datos de elementos del grupo del conjunto de datos relacionado con su objetivo.
- `objectiveSensitivity`: especifique el nivel de énfasis que la solución pone en su objetivo al generar recomendaciones. El nivel de sensibilidad de los objetivos configura la forma en que Amazon Personalize equilibra la recomendación de elementos en función de su objetivo frente a la relevancia a través de los datos de interacciones de elementos. El valor `objectiveSensitivity` puede ser `OFF`, `LOW`, `MEDIUM` o `HIGH`. Para obtener más información, consulte Equilibrio del énfasis y la relevancia del objetivo.

Utilice el siguiente código para crear una solución con un objetivo adicional con AWS SDK para Python (Boto3) o AWS SDK for Java 2.x.

Cuando la solución esté lista, cree una nueva versión de la solución (para ver un ejemplo de código, consulte Creación de una versión de la solución AWS (SDK de)). Una vez que haya creado una versión de la solución, podrá ver el rendimiento de la optimización con las métricas de la versión de la solución. Consulte Medición del rendimiento de optimización.

SDK for Python (Boto3)

Para crear una solución optimizada para un objetivo adicional, utilice el siguiente método `create_solution`. Reemplace `solution name`, `dataset group arn` y `recipe arn` por los suyos.

Para `optimizationObjective`, sustituya `COLUMN_NAME` por el nombre de la columna numérica de metadatos del conjunto de datos de elementos que esté relacionada con su objetivo. Para`objectiveSensitivity`, especifique OFF, LOW, MEDIUM o 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

Para crear una solución optimizada para un objetivo adicional, utilice el siguiente método `createPersonalizeSolution` y pase los siguientes parámetros: un cliente de servicio de Amazon Personalize, el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos, el nombre de la solución, el ARN de la receta, el atributo del elemento y el nivel de sensibilidad del objetivo.

```
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();
```

Cuaderno de Jupyter de ejemplo

Para ver un ejemplo de bloc de notas de Jupyter que muestra cómo crear una solución optimizada para metadatos de elementos adicionales basados en objetivos, consulte la carpeta objective_optimization del repositorio de muestras de Amazon Personalize. GitHub

Optimización de una solución con la configuración de eventos

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Si utiliza la receta Personalization-v2 o Personalized-Ranking-v2, puede optimizar una solución de Amazon Personalize con una configuración de eventos.

Con las recetas de recomendación de elementos, el objetivo principal de Amazon Personalize es predecir los artículos más relevantes para sus usuarios en función de los datos históricos y en

tiempo real de las interacciones con artículos. Sin embargo, la interacción puede incluir información adicional; por ejemplo, si un usuario ha hecho clic en un artículo determinado o si lo ha comprado. Puede registrarlo grabando el tipo del evento (Datos de valor de evento y tipo de evento). Al configurar una solución, puede hacer que asigne ponderaciones distintas a los diferentes tipos de evento de interacción. Por ejemplo, puede configurar una solución para dar más peso a los eventos de `purchase` que a los de `click`.

Para que una solución asigne diferentes ponderaciones a los distintos tipos de evento, al configurar los eventos de la solución, debe especificar los tipos de evento y sus ponderaciones correspondientes. Además, puede definir un umbral de valor de evento para excluir las interacciones que tengan un valor de evento inferior a ese umbral. Por ejemplo, si los datos EVENT_VALUE de los eventos con un EVENT_TYPE de ver son el porcentaje de un vídeo que ha visto un usuario, si establece el umbral del valor del evento en 0,5 y el tipo de evento en ver, Amazon Personalize entrena el modelo utilizando únicamente eventos de interacción ver con un EVENT_VALUE superior o igual a 0,5.

Las ponderaciones asociadas a los tipos de evento determinarán su importancia. Un tipo de evento con mayor ponderación hará que sea más probable que el modelo entrenado recomiende un artículo con el que se interactuaría con ese tipo de evento. Por ejemplo, si ha especificado "compra" con una ponderación mayor que "clic" y el modelo ha aprendido que, según su historial de interacciones, un usuario haría clic en el artículo B o compraría el artículo C, el modelo clasificará el artículo C en una posición más alta.

Para optimizar una solución con la configuración de eventos, cree una nueva solución con la receta User-Personalization-v2 o Personalized-Ranking-v2 y especifique una configuración de eventos. También puede actualizar una solución existente (Actualización de una solución para cambiar la configuración de entrenamiento automático) con una configuración de eventos.

Puede usar la consola de Amazon Personalize, AWS Command Line Interface (AWS CLI) o los SDK de AWS. Para obtener información sobre el uso de la consola de Amazon Personalize, consulte Creación de una solución (consola).

Temas

- Directrices y requisitos
- Medición del rendimiento con la configuración de ponderación de eventos
- Optimización de una solución (AWS CLI)
- Optimización de una solución (SDK de AWS)

Directrices y requisitos

Estas son las directrices y requisitos detallados de la configuración de eventos:

- Para configurar las ponderaciones de los distintos tipos de evento, su conjunto de datos de interacciones con artículos debe tener una columna EVENT_TYPE y, opcionalmente, una columna EVENT_VALUE.
- Puede especificar una lista de parámetros de evento en la configuración. Incluya todos los tipos de evento que desee tener en cuenta para la creación de la solución. Puede especificar un máximo de 10 tipos de evento diferentes.
- Puede especificar la ponderación del evento para cada tipo de evento. La ponderación del evento debe tener un valor entre 0,0 y 1,0. Solo importa la relación de las ponderaciones entre tipos de eventos. Por ejemplo, si se establece el tipo de evento "compra" con una ponderación de 0,3 y el tipo de evento "clic" con un ponderación de 0,1, tendrá el mismo resultado que si se establece el tipo de evento "compra" con una ponderación de 0,6 y "clic" con un ponderación de 0,2.
- Puede actualizar la configuración de eventos de una solución existente utilizando la operación de la API UpdateSolution.

Medición del rendimiento con la configuración de ponderación de eventos

Cuando crea una versión de una solución (entrena un modelo) con una configuración de eventos, Amazon Personalize genera una métrica `normalized_discounted_cumulative_gain_with_event_weights_at_k`. La puntuación de `normalized_discounted_cumulative_gain_with_event_weights_at_k` indica lo bien que funciona la versión de la solución teniendo en cuenta la ponderación que haya establecido para cada tipo de evento.

Esta métrica es similar a la ganancia acumulada descontada normalizada (NCDG) en K, pero se ponderará la recompensa por cada predicción correcta. Por el contrario, en el NDCG original en K, cada predicción correcta tendrá una ponderación de 1. Por ejemplo, si "compra" tiene una ponderación de 0,3 y "clic" una ponderación de 0,1, la recompensa por pronosticar correctamente la "compra" del artículo será 1,5, mientras que si se predice "clic", la recompensa será de 0,5.

Para obtener más información sobre la generación de métricas, consulte Evaluación de una versión de solución de Amazon Personalize con métricas.

Optimización de una solución (AWS CLI)

Puede optimizar la configuración de eventos con la receta User-Personalization-v2 o Personalized-Ranking-v2.

Para optimizar una solución con la configuración de eventos mediante la AWS CLI, cree una nueva solución y especifique los detalles de la configuración de eventos con la clave `eventsConfig` en el objeto `solutionConfig`. El evento `eventsConfig` tiene una clave de `eventParametersList` con la que puede especificar hasta 10 eventParameters. Cada `eventParameter` tiene los siguientes campos:

- eventType: especifique el tipo de evento que desea que se tenga en cuenta para la creación de la solución.
- eventValueThreshold: especifique el umbral del valor de evento. Para crear la solución, solo se tendrán en cuenta los eventos con un valor de evento superior o igual a este umbral.
- weight: especifique el peso de cada tipo de evento. Una ponderación más alta significa que el tipo de evento tiene mayor importancia para la solución creada.

A continuación, se muestra un ejemplo del comando de la AWS CLI create-solution. Reemplace `solution name`, `dataset group arn` y `recipe arn` por los suyos.

```
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}]}"
```

Cuando la solución esté lista, cree una nueva versión de la solución (para ver un ejemplo de comando, consulte Creación de una solución (AWS CLI)). Una vez que haya creado una versión de la solución, podrá ver el rendimiento de la optimización con las métricas de la versión de la solución. Consulte Medición del rendimiento de optimización.

Optimización de una solución (SDK de AWS)

Puede optimizar la configuración de eventos con la receta User-Personalization-v2 o Personalized-Ranking-v2.

Para optimizar una solución con la configuración de eventos mediante los SDK de AWS, cree una solución y especifique los detalles de la configuración de eventos con la clave `eventsConfig` en el objeto `solutionConfig`. El evento `eventsConfig` tiene una clave de `eventParametersList` con la que puede especificar hasta 10 `eventParameters`. Cada `eventParameter` tiene los siguientes campos:

- eventType: especifique el tipo de evento que desea que se tenga en cuenta para la creación de la solución.
- eventValueThreshold: especifique el umbral del valor de evento. Para crear la solución, solo se tendrán en cuenta los eventos con un valor de evento superior o igual a este umbral.
- weight: especifique el peso de cada tipo de evento. Una ponderación más alta significa que el tipo de evento tiene mayor importancia para la solución creada.

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 "";
```

Cuando la solución esté lista, cree una nueva versión de la solución (para ver un ejemplo de comando, consulte Creación de una solución (SDK de AWS)). Una vez que haya creado una versión de la solución, podrá ver el rendimiento de la optimización con las métricas de la versión de la solución. Consulte Medición del rendimiento de optimización.

Hiperparámetros y HPO

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Debe especificar hiperparámetros antes del entrenamiento para optimizar el modelo entrenado para su caso de uso concreto. Esto contrasta con los parámetros del modelo cuyos valores se determinan durante el proceso de entrenamiento.

Los hiperparámetros se especifican con la clave `algorithmHyperParameters` que forma parte del objeto SolutionConfig que se pasa a la operación CreateSolution.

A continuación se muestra una versión condensada de la solicitud `CreateSolution`. El ejemplo incluye el objeto `solutionConfig`. Puede utilizar `solutionConfig` para anular los parámetros predeterminados de una receta.

```
{
  "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"
        }
    },
  },
}
```

Las distintas recetas usan diferentes hiperparámetros. Para los hiperparámetros disponibles, consulte las recetas individuales en Elección de una receta.

Habilitación de optimización de hiperparámetros

La optimización de hiperparámetros (HPO), o su ajuste, es la tarea de elegir los hiperparámetros óptimos para alcanzar un objetivo de aprendizaje determinado. Los hiperparámetros óptimos se determinan mediante la ejecución de muchos trabajos de entrenamiento con diferentes valores de los rangos de posibilidades especificados.

Con la formación automática User-Personalization-v2y Personalized-Ranking-v2, si la activas, Amazon Personalize realiza automáticamente la HPO cada 90 días. Sin un entrenamiento automático, no se produce dicha optimización. Para todas las demás recetas, debe habilitar la optimización de hiperparámetros. Para usar HPO, establezca `performHP0` en `true` e incluya el objeto `hpoConfig`.

Los hiperparámetros pueden ser categóricos o continuos o tener valores enteros. El objeto `hpoConfig` tiene claves que corresponden a cada uno de estos tipos, donde se especifican los hiperparámetros y sus rangos. Debe proporcionar cada tipo en su solicitud, pero si una receta no tiene un parámetro de un tipo, puede dejarla vacía. Por ejemplo, User-Personalization no tiene un hiperparámetro ajustable de tipo continuo. Por tanto, para el `continousHyperParameterRange`, pasaría una matriz vacía.

El código siguiente muestra cómo crear una solución con la HPO habilitada mediante. AWS SDK para Python (Boto3) La solución del ejemplo usa la receta Receta User-Personalization y tiene el HPO establecido en `true`. El código proporciona un valor para `hidden_dimension` y `categoricalHyperParameterRanges` y `integerHyperParameterRanges`. El `continousHyperParameterRange` está vacío y el `hpoResourceConfig` establece el `maxNumberOfTrainingJobs` y `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"
          }
        }
    }
)
```

Para obtener más información acerca de HPO, consulte Ajuste de modelo automático.

Visualización de hiperparámetros

Puede ver los hiperparámetros de la solución llamando a la operación DescribeSolution. El siguiente ejemplo muestra un resultado `DescribeSolution`. Tras crear una versión de la solución (entrenando un modelo), también puede ver los hiperparámetros con la operación DescribeSolutionVersion.

```
{
  "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"
  }
}
```

Elección de los datos de interacciones de elementos utilizados para el entrenamiento

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Puede elegir los eventos de un conjunto de datos de interacciones de elementos que Amazon Personalize utiliza al crear una versión de la solución (entrenamiento de un modelo). La elección de los datos de interacciones de elementos antes del entrenamiento le permite usar solo un subconjunto relevante de los datos para el entrenamiento o eliminar ruido para entrenar un modelo más optimizado. Para obtener más información sobre los conjuntos de datos de interacciones de elemento, consulte Datos de interacciones de elementos.

> **Note**
>
> Si utilizas User-Personalization-v2 o Personalized-Ranking-v2, el coste de la formación se basa en los datos de interacciones entre los artículos antes de filtrarlos por tipo o valor del evento. Para obtener más información sobre el precio, consulte Precios de Amazon Personalize.

Puede elegir los datos de interacciones de elementos de la siguiente forma:

- Elija los registros según el tipo: al configurar una solución, si su conjunto de datos de interacciones de elementos incluye tipos de eventos en una columna EVENT_TYPE, puede especificar de manera opcional un tipo de evento para usarlo en el entrenamiento. Por ejemplo, si su conjunto de datos de interacciones incluye tipos de eventos compra, clic y ver, y desea que Amazon Personalize entrene el modelo solo para eventos ver, al configurar su solución, debe proporcionar visualización como el `event type` que Amazon Personalize debe usar en el entrenamiento.

  Si tienes varios tipos de eventos y utilizas la User-Personalization-v2 receta o la Personalized-Ranking-v2 receta, al configurar una solución personalizada puedes especificar distintos pesos para los distintos tipos. Por ejemplo, puede configurar una solución para dar más peso a los eventos de compra que a los de clic. Para obtener más información, consulte Optimización de una solución con la configuración de eventos.

  Si su conjunto de datos de interacciones de elementos tiene varios tipos de eventos en una columna EVENT_TYPE y no proporciona ningún tipo de evento al configurar la solución, Amazon Personalize utiliza todos los datos de interacciones de elementos para el entrenamiento con la misma ponderación, con independencia del tipo.
- Elija los registros según el tipo y el valor: al configurar una solución, si su conjunto de datos de interacciones de elementos incluye los campos EVENT_TYPE y EVENT_VALUE, puede establecer un valor específico como umbral para excluir registros del entrenamiento. Por ejemplo, si los datos EVENT_VALUE de los eventos con un EVENT_TYPE de ver son el porcentaje de un vídeo que ha visto un usuario, si establece el umbral del valor del evento en 0,5 y el tipo de evento en ver, Amazon Personalize entrena el modelo utilizando únicamente eventos de interacción ver con un EVENT_VALUE superior o igual a 0,5.

En el siguiente código se muestra cómo utilizar el AWS SDK para Python (Boto3) para crear una solución que utilice únicamente `watch` los eventos en los que el usuario haya visto más de la mitad del vídeo.

```
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'])
```

Clonación de una solución (consola)

Al crear una solución nueva, puede utilizar la consola de Amazon Personalize para clonar una solución. Al clonar una solución, puede utilizar la configuración de la solución existente como punto de partida, como la receta y los hiperparámetros, y realizar los cambios necesarios. Esto resulta útil si desea realizar un cambio en una solución, pero dejar inalteradas todas las demás propiedades. Por ejemplo, añadir una nueva columna de datos de entrenamiento a su conjunto de datos. En este caso, clonaría una solución, asignaría un nombre a la solución, cambiaría las columnas utilizadas durante el entrenamiento y dejaría todas las demás propiedades sin cambios.

Clonación de una solución

Para clonar una solución, elija la solución existente y la opción Clonar solución. A continuación, asigne un nombre a la nueva solución y modifique los campos correspondientes.

Para crear una solución

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. Elija Recursos personalizados y Soluciones.
4. Elija la solución que desea clonar.
5. Elija Acciones y Clonar solución.
6. Asigne un nombre a la nueva solución.

7. Realice los cambios que desee en los detalles de la solución y en la configuración avanzada. Amazon Personalize rellena previamente estos campos con valores de la solución existente. Para obtener más información acerca de cada campo, consulte Configuración de una solución personalizada en Amazon Personalize.

Actualización de una solución para cambiar la configuración de entrenamiento automático

Después de crear una solución, puede cambiar su configuración de entrenamiento automático y la configuración de sus eventos:

- Puede activar o desactivar el entrenamiento automático y cambiar la frecuencia del entrenamiento.
  - Si activa el entrenamiento automático, el primero comenzará una hora después de que se complete la actualización de la solución. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático.
  - Si modifica la frecuencia de entrenamiento de la solución, el programa de entrenamiento se restablece y una nueva versión de solución comienza el entrenamiento en una hora. La creación de la versión de solución continúa con la nueva frecuencia, donde el primer día es el día en que se actualiza la solución.
- Puede actualizar la configuración de eventos de las soluciones. Si la solución ya tiene especificada una configuración de eventos, la nueva configuración sustituirá a la original. Para obtener información sobre cómo optimizar una solución para una configuración de eventos, consulte Optimización de una solución con la configuración de eventos.

Puede actualizar una solución con la consola Amazon Personalize o AWS los SDK. AWS Command Line Interface Las actualizaciones de la solución pueden tardar unos minutos. Mientras la actualización esté en curso, puede crear versiones de solución, pero no puede eliminarla. Hasta que se complete la actualización, la solución usa la configuración anterior. Para obtener más información sobre el entramiento automático, consulte Configuración del entrenamiento automático.

Temas

- Actualización de una solución (consola)
- Actualización de una solución (AWS CLI)
- Actualizar una solución (AWS SDK)

Actualización de una solución (consola)

Para actualizar una solución en la consola, vaya hasta la solución, elija Actualizar y especifique la nueva configuración que desee utilizar.

Configuración de una solución

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En el panel de navegación, seleccione Recursos personalizados y elija Soluciones y recetas.
4. Seleccione la solución y elija Actualizar en la parte superior derecha.
5. En Entrenamiento automático, indique si la solución utiliza el entrenamiento automático. Si el entrenamiento automático está activado, puede cambiar la `Automatic training frequency`. La frecuencia de entrenamiento predeterminada es cada siete días.
6. Elija Actualizar solución. Puede encontrar el estado de la actualización de la solución en la página de detalles de la solución.

Actualización de una solución (AWS CLI)

Para actualizar una solución con AWS Command Line Interface, utilice el `update-solution` comando. Este comando utiliza la operación de la API UpdateSolution. El siguiente código muestra cómo actualizar una solución para utilizar el entrenamiento automático con una frecuencia de entrenamiento de cinco días. Para desactivar el entrenamiento automático, especifique `--no-perform-auto-training` y omita `solution-update-config`.

La frecuencia de entrenamiento predeterminada es cada siete días. La expresión debe tener el formato `rate(`*`value unit`*`)`. Para el valor, especifique un número comprendido entre 1 y 30. Para la unidad, especifique `day` o `days`.

```
aws personalize update-solution \
--solution-arn solution ARN \
--perform-auto-training \
--solution-update-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5
 days)\"}}"
```

Para obtener el estado de la actualización, utilice el comando `describe-solution` (que usa la operación de la API DescribeSolution) y busque el estado de la actualización en `latestSolutionUpdate`.

Actualizar una solución (AWS SDK)

Para actualizar una solución con los AWS SDK, utilice la operación UpdateSolution API. En el siguiente código se muestra cómo utilizar la solución AWS SDK para Python (Boto3) para que utilice el entrenamiento automático con una frecuencia de entrenamiento de 5 días. El código obtiene el estado de la actualización con la operación de la API DescribeSolution.

La frecuencia de entrenamiento predeterminada es cada siete días. La expresión debe tener el formato `rate(value unit)`. Para el valor, especifique un número comprendido entre 1 y 30. Para la unidad, especifique `day` o `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}")
```

Creación manual de una versión de solución

Después de completar Configuración de una solución personalizada en Amazon Personalize, podrá empezar a entrenar:

- Si la solución utiliza el entrenamiento automático, crea versiones de solución para usted con la frecuencia de entrenamiento que especifique. De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático para crear una nueva versión de solución cada siete días. Aún puede crear manualmente versiones de solución. Para obtener más información, consulte Configuración del entrenamiento automático.
- Si desactiva el entrenamiento automático para la solución o desea entrenarla manualmente, puede crear una versión de solución de forma manual. Una versión de la solución hace referencia a un modelo de machine learning entrenado. Puede crear una versión de la solución mediante la consola, AWS Command Line Interface (AWS CLI) o los SDK de AWS. Si la versión de la solución tiene un estado CREATE_PENDING o CREATE_IN_PROGRESS, puede utilizar la operación the section called “StopSolutionVersionCreation” para detener el proceso de creación de la versión de la solución. Consulte Detención de la creación de una versión de solución.

Si el entrenamiento no se completa debido a un error, no se le cobrará por él. Si la versión de la solución tiene un estado de CREATE_PENDING o CREATE_IN_PROGRESS, puede detener el proceso de creación de la versión de la solución. Para detener la creación de la versión de la solución, vaya a la página de detalles de la versión de la solución y elija Detener. Para obtener más información, consulte Detención de la creación de una versión de solución.

Temas

- Creación de una versión de la solución (consola)
- Creación de una versión de solución (AWS CLI)
- Creación de una versión de la solución AWS (SDK de)

Creación de una versión de la solución (consola)

Para crear manualmente una nueva versión de solución con la consola de Amazon Personalize, comience a entrenar desde la página de detalles de la solución.

Para crear una nueva versión de la solución

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Navegue hasta la página de grupos de conjuntos de datos y elija el grupo de conjuntos de datos con su nueva solución.
3. En el panel de navegación, en Recursos personalizados, elija Soluciones y recetas.

4. En la página Soluciones y recetas, elija la solución para la que desee crear una versión de la solución.
5. En la página de información general de la solución, elija Crear versión de la solución para empezar a entrenar un nuevo modelo.

En la página de detalles de la solución, puede realizar un seguimiento del progreso del entrenamiento en la sección Versiones de la solución. Cuando complete el entrenamiento, el estado será Activo y podrá evaluarlo mediante las métricas proporcionadas por Amazon Personalize. Para obtener más información, consulte Evaluación de una versión de solución de Amazon Personalize con métricas.

Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

- Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
- Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Creación de una versión de solución (AWS CLI)

Cuando su solución esté ACTIVA, para entrenar el modelo, ejecute el siguiente comando. Sustituya `solution arn` por el nombre de recurso de Amazon (ARN) de la solución de Configuración de una solución personalizada en Amazon Personalize.

```
aws personalize create-solution-version \
  --solution-arn solution arn
```

Se muestra el ARN de la versión de solución; por ejemplo:

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>"
}
```

Compruebe el estado del proceso de entrenamiento de la versión de solución utilizando el comando `describe-solution-version`. Proporcione el ARN de versión de solución que se devolvió en el paso anterior. Para obtener más información sobre la API, consulte DescribeSolutionVersion.

```
aws personalize describe-solution-version \
  --solution-version-arn solution version arn
```

Se muestran las propiedades de la versión de solución y el valor de `status` del entrenamiento. En un principio, el estado mostrado es CREATE PENDING, por ejemplo:

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

El entrenamiento se completa cuando el `status` es `ACTIVE` y puede evaluarlo utilizando las métricas proporcionadas por Amazon Personalize. Para obtener más información, consulte Evaluación de una versión de solución de Amazon Personalize con métricas. Si el entrenamiento no se completa debido a un error, no se le cobrará por él.

Si la versión de la solución tiene un estado CREATE_PENDING o CREATE_IN_PROGRESS, puede utilizar la operación StopSolutionVersionCreation para detener el proceso de creación de la versión de la solución. Consulte Detención de la creación de una versión de solución.

Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

- Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
- Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Creación de una versión de la solución AWS (SDK de)

Cuando la solución esté ACTIVA, utilice el siguiente código para crear una versión de la solución. Especifique el nombre de recurso de Amazon (ARN) de Configuración de una solución personalizada en Amazon Personalize. Utilice la operación DescribeSolutionVersion para recuperar el estado de la versión de la solución.

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();
```

Para comprobar el estado actual de la versión de la solución, llame a la operación DescribeSolutionVersion y pase el ARN de la versión de la solución que devuelve la operación `CreateSolutionVersion`. El entrenamiento se completa cuando el `status` es `ACTIVE` y puede evaluarlo utilizando las métricas proporcionadas por Amazon Personalize. Para obtener más información, consulte Evaluación de una versión de solución de Amazon Personalize con métricas. Si el entrenamiento no se completa debido a un error, no se le cobrará por él.

Si la versión de la solución tiene un estado CREATE_PENDING o CREATE_IN_PROGRESS, puede utilizar la operación StopSolutionVersionCreation para detener el proceso de creación de la versión de la solución. Consulte Detención de la creación de una versión de solución.

Cuando el estado de la versión de solución sea ACTIVE, podrá usarla para recibir recomendaciones. El uso de una versión de solución activa depende de cómo reciba las recomendaciones:

- Para obtener recomendaciones en tiempo real, implemente una versión de solución con el estado ACTIVE con una campaña de Amazon Personalize. Utilice la campaña para obtener recomendaciones para los usuarios. Consulte Implementación de una versión de solución de Amazon Personalize con una campaña.
- Para las recomendaciones por lotes, especifique una versión de solución con el estado ACTIVO al crear un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Detención de la creación de una versión de solución

Si la versión de solución tiene el estado CREATE_PENDING o CREATE_IN_PROGRESS, puede utilizar la consola de Amazon Personalize o la operación StopSolutionVersionCreation para dejar de detener la creación de la versión de solución (detener el entrenamiento de un modelo). No podrá reanudar la creación de una versión de solución una vez que se haya detenido. Se le facturarán los recursos utilizados hasta el momento en que se detuvo la creación de la versión de solución.

Si detiene la creación de una versión de solución, finaliza el entrenamiento del modelo, pero no se elimina la versión de solución. Aún puede ver los detalles de la versión de solución en la consola de Amazon Personalize y con la operación DescribeSolutionVersion.

Puede detener el proceso de creación de la versión de la solución con la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) o el AWS SDKs.

Temas

- Detención de la creación de una versión de solución (consola)
- Detención de la creación de una versión de solución (AWS CLI)
- Detención de la creación de una versión de solución (AWS SDKs)

Detención de la creación de una versión de solución (consola)

Si la versión de solución tiene el estado CREATE_PENDING o CREATE_IN_PROGRESS, puede detener la creación de una versión de solución (detener el entrenamiento de un modelo).

Para detener la creación de una versión de solución (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija el grupo de conjuntos de datos con la versión de solución que desee detener.
3. En el panel de navegación, elija Soluciones y recetas.
4. En la página Soluciones y recetas, elija la solución con la versión de solución que desee detener.
5. En Versiones de la solución, elija la versión de solución que desee detener.

6. En la página de detalles de la versión de solución, seleccione Detener la creación. Según el estado original de la versión de solución, el estado de la versión de solución cambia de la siguiente manera:

   - CREATE_PENDING cambia a CREATE_STOPPED.
   - CREATE_IN_PROGRESS cambia a CREATE_STOPPING y, a continuación, a CREATE_STOPPED.

Detención de la creación de una versión de solución (AWS CLI)

Si la versión de solución tiene el estado CREATE_PENDING o CREATE_IN_PROGRESS, puede detener la creación de una versión de solución (detener el entrenamiento de un modelo). Utilice el siguiente comando `stop-solution-version-creation` para dejar de crear la versión de solución con AWS CLI. Sustituya `solution version arn` por el nombre de recurso de Amazon (ARN) de la versión de solución que quiera detener. Se le facturarán los recursos utilizados hasta el momento en que se interrumpa la creación de la versión de solución.

```
aws personalize stop-solution-version-creation \
    --solution-version-arn solution version arn
```

Compruebe el estado del proceso de entrenamiento de la versión de solución utilizando el comando `describe-solution-version`.

```
aws personalize describe-solution-version \
    --solution-version-arn solution version arn
```

Según el estado original de la versión de solución, el estado de la versión de solución cambia de la siguiente manera:

- CREATE_PENDING cambia a CREATE_STOPPED.
- CREATE_IN_PROGRESS cambia a CREATE_STOPPING y, a continuación, a CREATE_STOPPED.

Detención de la creación de una versión de solución (AWS SDKs)

Si la versión de solución tiene el estado CREATE_PENDING o CREATE_IN_PROGRESS, puede detener la creación de una versión de solución (detener el entrenamiento de un modelo). En el

siguiente código se muestra cómo dejar de crear una versión de la solución con AWS SDK para Python (Boto3) o AWS SDK for Java 2.x. Se le facturarán los recursos utilizados hasta el momento en que se detenga la creación de la versión de solución.

SDK for Python (Boto3)

Utilice el siguiente método `stop_solution_version_creation` para detener la creación de una versión de solución. Sustituya `solution_version_arn` por el nombre de recurso de Amazon (ARN) de la versión de solución que quiera detener. El método utiliza la operación DescribeSolutionVersion para recuperar el estado de la versión de la solución.

```
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

Utilice el siguiente método `stopSolutionVersionCreation` para detener la creación de una versión de solución. Pase como parámetros un cliente de servicio de Amazon Personalize y el nombre de recurso de Amazon (ARN) de la versión de solución cuya creación desee detener. El siguiente código utiliza la operación DescribeSolutionVersion para recuperar el estado de la versión de solución.

```
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);
}
```

Según el estado original de la versión de solución, el estado de la versión de solución cambia de la siguiente manera:

- CREATE_PENDING cambia a CREATE_STOPPED.
- CREATE_IN_PROGRESS cambia a CREATE_STOPPING y, a continuación, a CREATE_STOPPED.

Evaluación de una versión de solución de Amazon Personalize con métricas

Puede evaluar el rendimiento de su versión de evaluación mediante métricas en línea y sin conexión. Las métricas en línea son los resultados empíricos que observa en las interacciones de sus usuarios con las recomendaciones en tiempo real. Por ejemplo, puede registrar la tasa de clics de sus usuarios mientras navegan por su catálogo. Usted es responsable de generar y registrar cualquier métrica en línea.

Las métricas sin conexión son las métricas que Amazon Personalize genera cuando entrena una versión de la solución. Con las métricas sin conexión, puede evaluar el rendimiento del modelo. Puede ver los efectos de modificar los hiperparámetros de una solución y puede comparar los resultados de los modelos entrenados con diferentes recetas con los mismos datos en el mismo grupo de conjuntos de datos.

Evite comparar las métricas de distintas versiones de solución entrenadas con datos diferentes. La diferencia en las métricas puede deberse a la diferencia en los datos y no al rendimiento del modelo. Por ejemplo, puede tener un grupo de conjuntos de datos con datos de evento `purchase` dispersos para cada usuario y otro con datos de eventos `view` sólidos. En función de métricas como `precision at K`, la versión de solución entrenada a partir de los datos de eventos de visualización podría parecer que tiene mejor rendimiento debido al mayor número de interacciones.

Para obtener las métricas de rendimiento, Amazon Personalize divide los datos de las interacciones de entrada en un conjunto de entrenamiento, un conjunto de prueba y, para PERSONALIZED_ACTIONS, un conjunto de validación. La división depende del tipo de receta que elija:

- En el caso de las recetas USER_SEGMENTATION, el conjunto de entrenamiento se compone del 80 % de los datos de interacciones de cada usuario y el conjunto de pruebas, del 20 % de los datos de interacciones de cada usuario.
- Para los demás tipos de recetas, el conjunto de entrenamiento consta del 90 % de los datos de sus usuarios y sus interacciones. El conjunto de pruebas está formado por el 10 % restante de los usuarios y sus datos de interacciones.

A continuación, Amazon Personalize crea la versión de la solución con el conjunto de entrenamiento. Una vez finalizado el entrenamiento, Amazon Personalize proporciona a la nueva versión de la solución el 90 % más antiguo de los datos de cada usuario del conjunto de pruebas como entrada. A continuación, Amazon Personalize calcula las métricas comparando las recomendaciones que genera la versión de la solución con las interacciones reales del 10 % más reciente de los datos de cada usuario del conjunto de pruebas.

Para generar una referencia con fines de comparación, le recomendamos usar la receta Popularity-Count, que recomienda los K elementos más populares.

Temas

- Recuperación de métricas de la versión de la solución
- Definiciones de métricas
- Ejemplo
- Recursos adicionales

Recuperación de métricas de la versión de la solución

Después de crear una versión de la solución, puede utilizar las métricas para evaluar su rendimiento. Puede recuperar las métricas de una versión de la solución con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) y AWSSDKs.

Temas

- Recuperación de métricas de la versión de la solución (consola)
- Recuperación de las métricas de la versión de la solución (AWS CLI)
- Recuperación de las métricas de la versión de la solución (AWSSDKs)

Recuperación de métricas de la versión de la solución (consola)

Para ver las métricas de los generadores de recomendaciones en la consola, vaya a la página de detalles de la versión de su solución.

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos personalizados.
3. En el panel de navegación, seleccione Recursos personalizados y elija Soluciones y recetas.
4. Elija su solución.
5. En Versiones de la solución, elija la versión de su solución para ver su página de detalles. Las métricas se muestran en la pestaña de Métricas de la versión de la solución, en el panel inferior. Para ver las definiciones de las métricas, consulte Definiciones de métricas.

   Ahora que ha evaluado su versión de la solución, puede crear una campaña implementando la versión de la solución con las mejores métricas par su caso de uso. Para obtener más información acerca cómo implementar una solución, consulte Implementación de una versión de solución de Amazon Personalize con una campaña.

Recuperación de las métricas de la versión de la solución (AWS CLI)

Las métricas de una versión de solución específica se recuperan llamando a la operación GetSolutionMetrics. En el siguiente código se muestra cómo recuperar las métricas con la AWS CLI.

```
personalize get-solution-metrics --solution-version-arn solution version ARN
```

A continuación se muestra un ejemplo de la salida de una versión de la solución creada mediante la receta User-Personalization con un objetivo de optimización adicional.

```
{
    "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
    }
}
```

Para obtener una explicación de cada métrica, consulte Definiciones de métricas. Ahora que ha evaluado su versión de la solución, puede crear una campaña implementando la versión de la solución con las mejores métricas par su caso de uso. Para obtener más información acerca cómo implementar una solución, consulte Implementación de una versión de solución de Amazon Personalize con una campaña.

Recuperación de las métricas de la versión de la solución (AWSSDKs)

Las métricas de una versión de solución específica se recuperan llamando a la operación GetSolutionMetrics. Utilice el siguiente código para recuperar las métricas.

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);
    }
}
```

A continuación se muestra un ejemplo de la salida de una versión de la solución creada mediante la receta User-Personalization con un objetivo de optimización adicional.

```
{
    "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
    }
}
```

Para obtener una explicación de cada métrica, consulte Definiciones de métricas. Ahora que ha evaluado su versión de la solución, puede crear una campaña implementando la versión de

la solución con las mejores métricas par su caso de uso. Para obtener más información acerca cómo implementar una solución, consulte Implementación de una versión de solución de Amazon Personalize con una campaña.

Definiciones de métricas

Las métricas que Amazon Personalize genera para las versiones de soluciones se describen a continuación con los siguientes términos:

- La recomendación relevante es una recomendación para un elemento con el que el usuario ha interactuado realmente. Estos elementos provienen del 10 % más reciente de los datos de interacciones de cada usuario del conjunto de pruebas.
- Laclasificación hace referencia a la posición de un elemento recomendado en la lista de recomendaciones. Se supone que la posición 1 (al principio de la lista) es la más importante para el usuario.

Para cada métrica, los números más altos (cercanos a 1) son mejores. Para profundizar aún más, consulte los recursos que se enumeran en Recursos adicionales.

coverage

El valor de cobertura indica la proporción de elementos (para recomendaciones de elementos), acciones (para recomendaciones de acciones) o usuarios (para recomendaciones de segmentos de usuarios) únicos que Amazon Personalize puede recomendar frente al número total de registros únicos de los conjuntos de datos.

Una puntuación de cobertura más elevada significa que Amazon Personalize recomienda más elementos de su catálogo, en lugar de los mismos registros de forma repetida. Las recetas que incluyen la exploración de elementos, como User-Personalization, tienen una mayor cobertura que las que no lo hacen, como Similar-Items.

rango recíproco medio en 25

Esta métrica indica la capacidad de un modelo para generar recomendaciones relevantes de elementos en la posición mejor clasificada.

Puede elegir un modelo con un rango recíproco medio alto de 25 si está generando resultados de búsqueda de elementos para un usuario y no espera que este elija un elemento que se encuentre más abajo en la lista. Por ejemplo, los usuarios suelen elegir la primera receta de

cocina en los resultados de búsqueda. Amazon Personalize no genera esta métrica para las recetas PERSONALIZED_ACTIONS ni USER_SEGMENTATION.

Amazon Personalize calcula esta métrica utilizando la puntuación de rango recíproco medio de las solicitudes de recomendaciones. Cada puntuación de rango recíproco se calcula de la siguiente manera: `1 / the rank of the highest item interacted with by the user`, donde el total de clasificaciones posibles es 25. Se ignoran otros elementos de menor rango con los que interactúa el usuario. Si el usuario eligió el primer elemento, la puntuación es 1. Si no elige ningún elemento, la puntuación es 0.

Por ejemplo, puede mostrar a tres usuarios diferentes 25 recomendaciones para cada uno:

- Si el usuario 1 hace clic en el elemento del rango 4 y en el elemento del rango 10, su puntuación de rango recíproco es de 1/4.
- Si el usuario 2 hace clic en un elemento del rango 2, en un elemento del rango 4 y en un elemento del rango 12, su puntuación de rango recíproco es de 1/2.
- Si el usuario 3 hace clic en un solo elemento del rango 6, su puntuación de rango recíproco es de 1/6.

El rango recíproco medio de todas las solicitudes de recomendaciones (en este caso, 3) se calcula de la siguiente manera: `(1/4 + 1/2 + 1/6) / 3 = .3056`.

ganancia acumulada descontada normalizada (NCDG) en K (5/10/25)

Esta métrica le indica lo bien que su modelo clasifica las recomendaciones de elementos o acciones, donde K es un tamaño de muestra de 5, 10 o 25 recomendaciones. Esta métrica es útil si lo que más le interesa es clasificar las recomendaciones más allá de la acción o el elemento mejor clasificado (para ello, consulte `mean reciprocal rank at 25`). Por ejemplo, la puntuación de `NDCG at 10` sería útil si tiene una aplicación que muestra hasta 10 películas en un carrusel a la vez.

Amazon Personalize calcula la NDCG asignando ponderación a las recomendaciones en función de su posición en la clasificación de cada usuario del conjunto de pruebas. Se descuenta cada recomendación (se le asigna una ponderación inferior) por un factor según su posición. La métrica final es el promedio de `NDCG at K` para todos los usuarios en el conjunto de pruebas. La `NDCG at K` supone que las recomendaciones que se encuentran más abajo en una lista son menos relevantes que las recomendaciones que están más arriba en la lista.

Amazon Personalize utiliza un factor de ponderación de `1/log(1 + position)`, donde el principio de la lista es la posición `1`.

precisión en K

Esta métrica indica la relevancia de las recomendaciones de su modelo, en función de un tamaño de muestra de K (de 5, 10 o 25) recomendaciones.

Amazon Personalize calcula esta métrica en función del número de recomendaciones relevantes de las K recomendaciones principales para cada usuario de las pruebas, dividido por K, donde K es 5, 10 o 25. La métrica final es la media de todos los usuarios en el conjunto de pruebas.

Por ejemplo, si recomiendas 10 elementos a un usuario y el usuario interactúa con 3 de ellos, la precisión en K es de 3 elementos predichos correctamente divididos por el total de 10 elementos recomendados: `3 / 10 = .30`.

Esta métrica recompensa la recomendación precisa de los elementos relevantes. Cuanto más cerca esté la puntuación a uno, más preciso será el modelo.

precisión

Si entrenas una versión de la solución con la Next-Best-Action receta, Amazon Personalize genera una `precision` métrica en lugar de`precision at K`. Esta métrica le indica cómo de bueno es su modelo para predecir las acciones que los usuarios realizarán realmente.

Para calcular `precision`, para cada acción del conjunto de datos, Amazon Personalize divide el número de usuarios que se predijo correctamente que realizaría la acción por el número total de veces que se recomendó la acción. A continuación, Amazon Personalize calcula el promedio de todas las acciones en el conjunto de datos.

Por ejemplo, si se recomendó una acción a 100 usuarios, de los cuales 60 la realizaron y 40 no, el valor `precision` de la acción es `60 / 100 = .60`. A continuación, Amazon Personalize aplica este cálculo a todas las acciones y devuelve el promedio.

Esta métrica recompensa la recomendación precisa de acciones relevantes. Cuanto más cerca esté la puntuación a uno, más preciso será el modelo.

media_recompensas_en_k

Cuando crea una versión de solución (entrena un modelo) para una solución con un objetivo de optimización, Amazon Personalize genera una métrica `average_rewards_at_k`. La puntuación de `average_rewards_at_k` indica el rendimiento de la versión de la solución en cuanto al logro de su objetivo. Para calcular esta métrica, Amazon Personalize calcula las recompensas para cada usuario de la siguiente manera:

```
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
```

El `average_rewards_at_k` final es el promedio de todos los valores `rewards_per_user` normalizados para que sea un valor decimal menor o igual a 1 y mayor que 0. Cuanto más se acerque el valor a 1, más beneficios de media por usuario se pueden esperar de las recomendaciones.

Por ejemplo, si su objetivo es maximizar los ingresos de los clics, Amazon Personalize calcula la puntuación de cada usuario dividiendo los ingresos totales generados por los elementos en los que el usuario hizo clic de sus 25 recomendaciones más caras entre los ingresos de todos los elementos recomendados en los que hizo clic el usuario. Amazon Personalize devuelve entonces un promedio normalizado de todas las puntuaciones de los usuarios. Cuanto más cerca esté `average_rewards_at_k` de 1, más ingresos en promedio se pueden esperar por usuario a partir de las recomendaciones.

Para obtener más información, consulte Optimización de una solución para un objetivo adicional.

normalized_discounted_cumulative_gain_with_event_weights_at_k

Cuando crea una versión de una solución (entrena un modelo) con una configuración de eventos, Amazon Personalize genera una métrica `normalized_discounted_cumulative_gain_with_event_weights_at_k`. La puntuación de `normalized_discounted_cumulative_gain_with_event_weights_at_k` indica lo bien que funciona la versión de la solución teniendo en cuenta la ponderación que haya establecido para cada tipo de evento.

Esta métrica es similar a la ganancia acumulada descontada normalizada (NCDG) en K, pero se ponderará la recompensa por cada predicción correcta. Por el contrario, en el NDCG original en K, cada predicción correcta tendrá una ponderación de 1. Por ejemplo, si "compra" tiene una ponderación de 0,3 y "clic" una ponderación de 0,1, la recompensa por pronosticar correctamente la "compra" del artículo será 1,5, mientras que si se predice "clic", la recompensa será de 0,5.

Para obtener más información, consulte Optimización de una solución con la configuración de eventos.

precisión de predicción de tendencias

Si entrenó la versión de la solución con la receta Trending-Now, la tasa de aumento de la popularidad de los elementos recomendados por el modelo. Cuanto mayor sea la precisión de

la predicción de las tendencias (cuanto más cerca de 1), mejor será el modelo para identificar correctamente los elementos populares.

Para calcular la aceleración de la popularidad, Amazon Personalize divide la tasa de aumento de la popularidad de todos los elementos recomendados entre el aumento total de popularidad de los 25 elementos más populares. Estos elementos provienen de las interacciones reales en el conjunto de pruebas.

En función de la distribución de los datos y de la frecuencia de detección de tendencias que elija, el valor de la precisión de la predicción de tendencias puede ser 0,0.

acierto (acierto en K)

Si entrenó la versión de la solución con una receta USER_SEGMENTATION, el número medio de usuarios entre los K más relevantes previstos que coinciden con los usuarios reales. Los usuarios reales son los usuarios que realmente interactuaron con los elementos del conjunto de prueba. K corresponde al 1 % superior de los usuarios más relevantes. Cuanto mayor sea el valor, más precisas serán las predicciones.

recuperación (recuperación en K)

Si entrenó la versión de la solución con una receta USER_SEGMENTATION, el porcentaje medio de usuarios previstos entre los K más relevantes previstos que coinciden con los usuarios reales. Los usuarios reales son los usuarios que realmente interactuaron con los elementos del conjunto de prueba. K corresponde al 1 % superior de los usuarios más relevantes. Cuanto mayor sea el valor, más precisas serán las predicciones.

exhaustividad

Si entrena una versión de la solución con la Next-Best-Action receta, esta métrica le indica qué tan buena es la versión de la solución a la hora de descubrir las acciones con las que interactuarán los usuarios.

Para calcular `recall`, para cada acción del conjunto de datos, Amazon Personalize divide el número de usuarios que se predijo correctamente que realizaría la acción por el número total de usuarios que realmente ha realizado la acción en el conjunto de prueba. A continuación, Amazon Personalize calcula el promedio de todas las acciones en el conjunto de datos.

Por ejemplo, si 100 usuarios realizan una acción en el conjunto de prueba y Amazon Personalize predijo que 50 de estos usuarios la realizarían, el valor `recall` de la acción es `50 / 100 = .50`. A continuación, Amazon Personalize aplica este cálculo a todas las acciones y devuelve el promedio.

Área bajo la curva (AUC)

Si ha entrenado la versión de la solución con una receta PERSONALIZED_ACTIONS, esta es el área situada bajo la curva de característica operativa del receptor para la versión de la solución. Esta métrica indica lo bien o no que funciona la versión de la solución a la hora de identificar correctamente las acciones que realizarán los usuarios.

La curva de característica operativa del receptor traza el rendimiento de la versión de la solución. Representa gráficamente las tasas de positivos reales (acciones que se predijeron correctamente como relevantes) y de falsos positivos (acciones que se predijeron incorrectamente como relevantes) en distintos valores de umbral. El área bajo la curva (AUC) es una puntuación que resume el rendimiento de la versión de la solución en función de su curva.

El valor de AUC de una versión de solución puede estar entre 0 y 1. Cuanto más se acerque a 1, mejor será el modelo a la hora de predecir las acciones relevantes para los usuarios.

Popularidad del artículo

Si has entrenado la versión de la solución con la `aws-semantic-similarity` receta, esta métrica mide el interés que ha recibido un artículo (me gusta, visitas, contenido compartido, etc.) en relación con otros artículos de tu catálogo. Cuanto más alta sea la puntuación de popularidad del artículo (cuanto más cerca de 1), más interacciones de los usuarios acumulará el artículo en comparación con otros artículos.

Para calcular la popularidad de un artículo, Amazon Personalize cuenta el número total de usuarios únicos que han interactuado con cada artículo (a través de eventos como visitas, clics, compras, etc.). Luego, el recuento bruto de interacciones de cada artículo se normaliza en todos los elementos de tu conjunto de datos, lo que da como resultado puntuaciones entre 0 y 1. Una puntuación más cercana a 0 indica una participación baja, mientras que una puntuación más cercana a 1 indica una participación media alta con los artículos del catálogo.

Frescura del artículo

Si has entrenado la versión de la solución con la `aws-semantic-similarity` receta, esta métrica mide la fecha en que se agregó un artículo a tu catálogo en relación con otros artículos. Cuanto más alta sea la puntuación de frescura del artículo (cuanto más cerca de 1), más recientemente se creó el artículo.

Para calcular la frescura de los artículos, Amazon Personalize determina la antigüedad de cada artículo mediante el CREATION_TIMESTAMP del conjunto de datos de tus artículos. La edad

se calcula como la diferencia entre el tiempo de entrenamiento y la marca temporal de creación del artículo. Luego, estas edades se normalizan en todos los objetos, y los más nuevos reciben puntuaciones cercanas a 1. Una puntuación más cercana a 0 indica más artículos antiguos, mientras que una puntuación más cercana a 1 indica que hay más artículos nuevos en promedio en el catálogo.

Similitud de artículos

Si has entrenado la versión de la solución con la `aws-semantic-similarity` receta, esta métrica mide la relación semántica de los elementos recomendados con los elementos de entrada en función de su contenido, atributos y metadatos. Cuanto más alta sea la puntuación de similitud de los elementos (cuanto más se acerque a 1), más coincidirán los elementos recomendados con las características de los artículos de referencia.

Para calcular la similitud de los artículos, Amazon Personalize calcula las puntuaciones de similitud entre cada artículo recomendado y los elementos de entrada mediante incrustaciones semánticas derivadas de los metadatos y atributos del artículo. Una puntuación cercana a 0 indica una baja similitud semántica entre los artículos del catálogo en promedio, mientras que una puntuación cercana a 1 indica una alta similitud semántica entre los artículos del catálogo en promedio. Esta métrica le ayuda a comprender qué tan bien el modelo capta las relaciones semánticas en sus recomendaciones.

Ejemplo

A continuación, se muestra un ejemplo sencillo para una versión de solución que produce una lista de recomendaciones para un usuario específico. La segunda y la quinta recomendación coinciden con los registros de los datos de prueba de este usuario. Estas son las recomendaciones pertinentes. Si `K` se establece en `5`, se generan las siguientes métricas para el usuario.

reciprocal_rank

Cálculo: 1/2

Resultado: 0,5000

normalized_discounted_cumulative_gain_at_5

Cálculo: (1/log(1 + 2) + 1/log(1 + 5)) / (1/log(1 + 1) + 1/log(1 + 2))

Resultado: 0,6241

precision_at_5

Cálculo: 2/5

Resultado: 0,4000

Recursos adicionales

Para obtener información sobre cómo evaluar una versión de solución con A/B pruebas, consulte Uso de A/B las pruebas para medir la eficacia de las recomendaciones generadas por Amazon Personalize. Para profundizar en los diferentes tipos de métricas para los sistemas de recomendaciones, consulte los siguientes recursos externos:

- MRR frente a MAP frente a NDCG: métricas de evaluación con conocimiento de la clasificación y cuándo utilizarlas
- Ganancia acumulada descontada: las métricas de clasificación que debe conocer
- Recuperación y precisión en k para sistemas de generadores de recomendaciones
- Clasificación de métricas de evaluación para sistemas de generadores de recomendaciones
- Característica operativa del receptor

Implementación de una versión de solución de Amazon Personalize con una campaña

Para obtener recomendaciones en tiempo real con recursos personalizados, una vez finalizada Creación manual de una versión de solución, estará listo para implementar la versión de la solución con una campaña.

Una campaña implementa una versión de la solución (modelo entrenado) con una capacidad de transacción aprovisionada para generar recomendaciones en tiempo real. Después de crear una campaña, puede utilizar las operaciones de la API de GetRecommendations o GetPersonalizedRanking para obtener recomendaciones. Si recibe recomendaciones de elementos o de segmentos de usuarios por lotes, no es necesario que cree una campaña. Para obtener más información, consulte Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes.

Al crear una campaña, puede configurar lo siguiente:

- Puede configurar la campaña para que se actualice automáticamente y utilice la versión de solución más reciente. Para obtener más información consulte () Actualizaciones de campaña automáticas.
- Puede habilitar los metadatos de elemento en las recomendaciones. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.
- Puede especificar el número mínimo de transacciones aprovisionadas por segundo para la campaña. Este es el rendimiento de transacciones de línea base para la campaña aprovisionada por Amazon Personalize. Establece el cargo de facturación mínimo de la campaña mientras esté activa. Para obtener más información, consulte Transacciones mínimas aprovisionadas por segundo y escalado automático.

Puede crear una campaña con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Si desea cambiar la configuración de una campaña existente, por ejemplo, habilitar los metadatos en las recomendaciones, debe actualizar la campaña. Para obtener más información, consulte Actualización de la configuración de una campaña de Amazon Personalize.

Generará costos de campaña mientras la campaña esté activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener información sobre los costos de campaña, consulte Precios de Amazon Personalize.

Temas

- Actualizaciones de campaña automáticas
- Transacciones mínimas aprovisionadas por segundo y escalado automático
- Metadatos de elemento en las recomendaciones
- Crear una campaña (consola)
- Creación de una campaña (AWS CLI)
- Creación de una campaña (SDK de AWS)

Actualizaciones de campaña automáticas

Al crear una campaña, puede habilitar las actualizaciones de campaña automáticas. Con las actualizaciones automáticas, la campaña se actualiza automáticamente para implementar la última versión de solución entrenada de forma automática o manual. Esto le facilita la actualización de la campaña.

Por ejemplo, si la solución usa el entrenamiento automático para crear una nueva versión de solución cada siete días, la campaña se actualizará automáticamente para usar la última versión de solución en cada entrenamiento semanal. Si no utiliza las actualizaciones de campaña automáticas, debe actualizarla manualmente para implementar el modelo entrenado más reciente.

- Para habilitar las actualizaciones de campaña automáticas al crear una campaña con la consola de Amazon Personalize, elija Actualizar automáticamente para usar la última versión de solución de la solución en los Detalles de la campaña. Puede encontrar la marca horaria de la última actualización en la página de detalles de la campaña.

  Para obtener más información, consulte Crear una campaña (consola).
- Para habilitar las actualizaciones de campaña automáticas cuando utilice la operación de la API CreateCampaign, especifique el formato `SolutionArn/$LATEST` del nombre de recurso de Amazon (ARN) de la solución para el parámetro `SolutionVersionArn`. En `campaignConfig`, establezca `enableMetadataWithRecommendations` en `true`.

  Para obtener la marca temporal de la última actualización de campaña, puede utilizar la operación de la API DescribeCampaign y comprobar los detalles de `latestCampaignUpdate` en la respuesta.

  Para ver muestras de código que presentan cómo activar las actualizaciones automáticas, consulte Creación de una campaña (AWS CLI) o Creación de una campaña (SDK de AWS).

Transacciones mínimas aprovisionadas por segundo y escalado automático

> **Important**
>
> Un valor elevado de `minProvisionedTPS` aumentará los costos. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realice un seguimiento de su uso mediante las métricas de Amazon CloudWatch y aumente el valor de `minProvisionedTPS` según sea necesario.

Al crear una campaña de Amazon Personalize, puede especificar el número mínimo de transacciones aprovisionadas por segundo (`minProvisionedTPS`) para la campaña. Este es el rendimiento de transacciones de línea base para la campaña aprovisionada por Amazon Personalize. Establece el cargo de facturación mínimo de la campaña mientras esté activa.

Una transacción es una solicitud `GetRecommendations` o `GetPersonalizedRanking` simple. El valor predeterminado de `minProvisionedTPS` es 1.

Si su TPS aumenta más de `minProvisionedTPS`, Amazon Personalize escala automáticamente la capacidad aprovisionada hacia arriba y hacia abajo, pero nunca por debajo de `minProvisionedTPS`. Se produce un breve retraso durante el aumento de la capacidad que podría provocar la pérdida de transacciones. Cuando se reduce el tráfico, la capacidad vuelve al valor de `minProvisionedTPS`.

Se le cobrará el valor de TPS aprovisionado mínimo o, si las solicitudes superan el valor de `minProvisionedTPS`, el TPS real. El TPS real es el número total de solicitudes de recomendación que realiza. Recomendamos empezar con un nivel bajo de `minProvisionedTPS`, realizar un seguimiento del uso mediante las métricas de Amazon CloudWatch y, a continuación, aumentar el `minProvisionedTPS` según sea necesario.

Para obtener más información sobre los costos de las campañas, consulte Precios de Amazon Personalize.

Metadatos de elemento en las recomendaciones

> **Important**
>
> Si usa la receta User-Personalization-v2 o Personalized-Ranking-v2, no generará costos adicionales por los metadatos. Para el resto de las recetas y todos los casos de uso de dominio, se generan costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Cuando reciba recomendaciones, puede hacer que Amazon Personalize incluya metadatos de elemento en los resultados de las recomendaciones. En la solicitud, puede elegir las columnas del conjunto de datos de elementos que se incluirán. Amazon Personalize devuelve estos datos para cada elemento en la respuesta a la recomendación.

Puede usar metadatos para enriquecer las recomendaciones de la interfaz de usuario, por ejemplo, agregar los géneros de las películas a los carruseles. También puede usarlos para evaluar la calidad de las recomendaciones de manera visual. Si usa IA generativa en la aplicación, puede conectar los metadatos a las indicaciones de la IA a fin de generar contenido más relevante. Para obtener más información sobre el uso de Amazon Personalize con la IA generativa, consulte Amazon Personalize y la IA generativa.

Habilitación de metadatos

Para agregar metadatos a las recomendaciones, debe tener un conjunto de datos de elementos con una columna de metadatos. No es necesario usar los metadatos en el entrenamiento. Para obtener información sobre cómo crear un conjunto de datos, consulte Creación de un esquema y un conjunto de datos. Para obtener información sobre la actualización de datos, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Si utiliza la receta User-Personalization-v2 o Personalized-Ranking-v2, las nuevas campañas tienen automáticamente la opción de incluir los metadatos de elemento con los resultados de las recomendaciones. No tiene que activar manualmente los metadatos de la campaña. Para el resto de las recetas y casos de uso de dominio, debes habilitar la opción de metadatos:

- Para habilitar los metadatos con la consola de Amazon Personalize, al crear la campaña, seleccione Devolver metadatos de los elementos en los resultados de las recomendaciones en Detalles de la campaña. Para obtener más información, consulte Crear una campaña (consola).
- Para habilitar los metadatos con los SDK de AWS o la AWS CLI, use la operación de API CreateCampaign y en `campaignConfig` establezca `enableMetadataWithRecommendations` en `true`. Para obtener más información, consulte Creación de una campaña (AWS CLI) o Creación de una campaña (SDK de AWS).

Crear una campaña (consola)

> **Important**
>
> Generará costos de campaña mientras la campaña esté activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener información sobre los costos de campaña, consulte Precios de Amazon Personalize.

Cuando el estado de la versión de la solución sea Activo, ya puede implementarla con una campaña de Amazon Personalize.

Para crear una campaña (consola)

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Elija el grupo de conjuntos de datos con la versión de solución que desea implementar.

3. En el panel de navegación, en Recursos personalizados, elija Campañas.
4. En la pestaña Campañas, elija Crear campaña.
5. En la página Crear nueva campaña, en los Detalles de la campaña, proporcione la siguiente información:

   - Nombre de la campaña: escriba el nombre de la campaña. El texto que escriba aquí aparecerá en el panel Campaña y en la página de detalles.
   - Solución: elija la solución que acaba de crear.
   - Actualizar automáticamente para usar la última versión de solución de la solución: elija esta opción para que la campaña use automáticamente la última versión de solución activa. Si no elige esta opción, debe actualizar manualmente la campaña cada vez que desee implementar una nueva versión de solución. Para obtener más información, consulte Actualizaciones de campaña automáticas.
   - ID de versión de solución: si no utiliza las actualizaciones de campaña automáticas para usar la versión de solución más reciente, elija el ID de la versión de solución que desea implementar.
   - Mínimo de transacciones aprovisionadas por segundo (denominadas minProvisionedTPS en las API): defina el mínimo de transacciones aprovisionadas por segundo que admite Amazon Personalize. Un valor elevado aumentará los gastos. Le recomendamos que comience con 1 (el valor predeterminado). Realice un seguimiento del uso mediante las métricas de Amazon CloudWatch y aumente el valor de `minProvisionedTPS` según sea necesario. Para obtener más información, consulte Transacciones mínimas aprovisionadas por segundo y escalado automático.
   - Devolver metadatos de elementos en los resultados de recomendaciones: elija esta opción si desea incluir los metadatos en los resultados de recomendaciones. Si está habilitada esta opción, puede especificar las columnas del conjunto de datos de elementos cuando reciba recomendaciones. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.
6. Si ha utilizado la receta User-Personalization, en Configuración de la campaña, si lo desea, puede especificar valores para Ponderación de exploración y Límite de antigüedad de los elementos de exploración. Para obtener más información, consulte User-Personalization.
7. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
8. Elija Crear una campaña.

9. En la página de detalles de la campaña, cuando el estado de la campaña es Activo, puede utilizarla para obtener recomendaciones y registrar las impresiones. Para obtener más información, consulte Obtención de recomendaciones de Amazon Personalize.

   La campaña estará lista cuando su estado sea ACTIVO. Si vuelve a entrenar la versión de solución o desea cambiar la configuración de la campaña, debe actualizarla. Para obtener más información, consulte Actualización de la configuración de una campaña de Amazon Personalize.

Creación de una campaña (AWS CLI)

> **Important**
>
> Generará costos de campaña mientras la campaña esté activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener información sobre los costos de campaña, consulte Precios de Amazon Personalize.

Cuando el estado de la versión de solución sea Activo, podrá implementarla con una campaña de Amazon Personalize. Para crear una campaña con la AWS CLI, utilice el comando `create-campaign`.

En la siguiente muestra de código se presenta cómo crear una campaña. Implementa la última versión de solución de una solución que utiliza la receta User-Personalization. La campaña que crea se actualiza automáticamente para usar futuras versiones de solución. El código utiliza la siguiente configuración:

- Configura la campaña para que se actualice automáticamente y utilice la versión de solución más reciente: `solution-version-arn` está en formato *`solution ARN`*`/$LATEST` y `syncWithLatestSolutionVersion` es `True`. Para utilizar el código, reemplace `solution ARN` por el nombre de recurso de Amazon (ARN) de la solución.

  Para deshabilitar el valor automático de `syncWithLatestSolutionVersion`, especifique únicamente el ARN de la versión de solución (sin `/$LATEST`) y establezca `syncWithLatestSolutionVersion` en `False`.
- Establece la opción `enableMetadataWithRecommendations` en `True`. Esto permite que una opción de solicitud de recomendación incluya los metadatos de elemento del conjunto de datos de

elementos en los resultados de las recomendaciones. Para deshabilitar esta opción, establézcala a `False`. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.

- Establece `min-provisioned-tps` en 1 (el valor predeterminado). Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realice un seguimiento del uso mediante las métricas de Amazon CloudWatch y aumente el valor de `minProvisionedTPS` según sea necesario. Para obtener más información, consulte Transacciones mínimas aprovisionadas por segundo y escalado automático.

Para obtener una lista completa de todos los parámetros, consulte CreateCampaign.

```
aws personalize create-campaign \
--name campaign-name \
--solution-version-arn solution-arn/\$LATEST \
--min-provisioned-tps 1 \
--campaign-config "{"\"syncWithLatestSolutionVersion"\": "true",
 "\"enableMetadataWithRecommendations"\": "true"}"
```

La campaña estará lista cuando su estado sea ACTIVO. Para obtener el estado actual de la campaña, llame a DescribeCampaign y compruebe que el valor de la propiedad `status` es `ACTIVE`.

Si vuelve a entrenar la versión de solución y la campaña no se actualiza automáticamente para usar la última versión de solución, o si desea cambiar la configuración de la campaña, debe actualizarla. Para obtener más información, consulte Actualización de la configuración de una campaña de Amazon Personalize.

Amazon Personalize le proporciona operaciones para administrar las campañas como, por ejemplo, ListCampaigns para obtener una lista de las campañas que ha creado. Puede eliminar una campaña llamando a DeleteCampaign. Si elimina una campaña, las versiones de solución que forman parte de la campaña no se eliminan.

Una vez que haya creado la campaña, puede utilizarla para realizar recomendaciones. Para obtener más información, consulte Obtención de recomendaciones de Amazon Personalize.

Creación de una campaña (SDK de AWS)

> **Important**
>
> Generará costos de campaña mientras la campaña esté activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener información sobre los costos de campaña, consulte Precios de Amazon Personalize.

Cuando el estado de la versión de solución sea Activo, podrá implementarla con una campaña de Amazon Personalize. Para crear una campaña con los SDK de AWS, utilice la operación de la API CreateCampaign.

En la siguiente muestra de código se presenta cómo crear una campaña. El código implementa la última versión de solución de una solución que utiliza la receta User-Personalization. La campaña que crea se actualiza automáticamente para usar futuras versiones de solución. El código utiliza la siguiente configuración:

- Configura la campaña para que se actualice automáticamente y utilice la versión de solución más reciente: `solutionVersionArn` está en formato *`solution ARN`*`/$LATEST` y `syncWithLatestSolutionVersion` es `True`. Para usar el código, reemplace `solution ARN` por el nombre de recurso de Amazon (ARN) de la versión de solución.

  Para deshabilitar el valor automático de `syncWithLatestSolutionVersion`, especifique únicamente el ARN de la versión de solución (sin `/$LATEST`) y establezca `syncWithLatestSolutionVersion` en `False`.
- Establece la opción `enableMetadataWithRecommendations` en `True`. Esto permite que una opción de solicitud de recomendación incluya los metadatos de elemento del conjunto de datos de elementos en los resultados de las recomendaciones. Para deshabilitar esta opción, establézcala a `False`. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.
- Establece `minProvisionedTPS` en 1 (el valor predeterminado). Le recomendamos que comience con 1 para `minProvisionedTPS` (el valor predeterminado). Realice un seguimiento del uso mediante las métricas de Amazon CloudWatch y aumente el valor de `minProvisionedTPS` según sea necesario. Para obtener más información, consulte Transacciones mínimas aprovisionadas por segundo y escalado automático.

Para obtener una lista completa de todos los parámetros, consulte CreateCampaign.

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 campaña estará lista cuando su estado sea ACTIVO. Para obtener el estado actual, llame a DescribeCampaign y compruebe que el valor del campo `status` es `ACTIVE`.

Si vuelve a entrenar manualmente la versión de solución o si desea cambiar la configuración de la campaña, debe actualizarla. Para obtener más información, consulte Actualización de la configuración de una campaña de Amazon Personalize.

Amazon Personalize le proporciona operaciones para administrar las campañas como, por ejemplo, ListCampaigns para obtener una lista de las campañas que ha creado. Puede eliminar una campaña llamando a DeleteCampaign. Si elimina una campaña, las versiones de solución que forman parte de la campaña no se eliminan.

Una vez que haya creado su campaña, utilícela para realizar recomendaciones. Para obtener más información, consulte Obtención de recomendaciones de Amazon Personalize.

Actualización de la configuración de una campaña de Amazon Personalize

Para cambiar el TPS mínimo aprovisionado de la campaña, implementar de forma manual una nueva versión de solución o modificar la configuración de una campaña (por ejemplo, activar la opción de incluir metadatos en las recomendaciones), debe actualizar la campaña de modo manual.

Lo que se indica a continuación no requiere una actualización manual de la campaña:

- Si la campaña usa actualizaciones de campaña automáticas, no tiene que actualizarla para implementar la última versión de solución configurada entrenada de forma automática o manual. Para obtener más información, consulte Actualizaciones de campaña automáticas.

- Con User-Personalization-v2 User-Personalization Next-Best-Action, o Amazon Personalize actualiza automáticamente la última versión de la solución cada dos horas para incluir nuevos elementos o acciones en las recomendaciones. La campaña utiliza automáticamente la versión actualizada de la solución.

Actualizas manualmente una campaña con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS los SDK.

> **Note**
>
> Para actualizar una campaña con el fin de incluir metadatos en las recomendaciones, debe tener un conjunto de datos de elementos con una columna de metadatos. No es necesario usar los metadatos en el entrenamiento.
> Si tu campaña implementó anteriormente una versión de la solución que utilizaba User-Personalization-v2 o Personalized-Ranking-v2, y vas a cambiar a una versión anterior de las recetas, la opción de incluir metadatos está desactivada de forma predeterminada. Puede activarla al actualizar la campaña. Para obtener más información, consulte Metadatos de elemento en las recomendaciones.

Temas

- Actualización de una campaña (consola)
- Actualizar una campaña (AWS CLI)
- Actualizar una campaña (AWS SDK)

Actualización de una campaña (consola)

Para implementar una versión de la solución reentrenada manualmente o realizar cambios en la configuración de su campaña, debe actualizar su campaña.

Para actualizar una campaña (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija el grupo de conjuntos de datos con la campaña que desea actualizar.
3. En el panel de navegación, seleccione Campañas.

4. En la página Campañas, elija la campaña que desea actualizar.
5. En la página de detalles de la campaña, seleccione Actualizar.
6. En la página Actualizar campaña, realice los cambios. Por ejemplo, si va a implementar una versión reentrada de la solución, en el campo ID de la versión de la solución, elija el número de identificación de la nueva versión de la solución.
7. Elija Actualizar. Amazon Personalize actualiza la campaña para usar la nueva versión de la solución y cualquier configuración modificada.

Actualizar una campaña (AWS CLI)

Para implementar una nueva versión de la solución, cambiar el TPS mínimo aprovisionado de la campaña o cambiar la configuración de la campaña, debe actualizarla. Use el siguiente comando `update-campaign` para actualizar una campaña y usar una nueva versión de la solución con AWS CLI.

Reemplace `campaign arn` por el nombre de recurso de Amazon (ARN) de la campaña que desea actualizar. Reemplace `new solution version arn` por la versión de la solución que desea implementar.

```
aws personalize update-campaign \
--campaign-arn campaign arn \
--solution-version-arn new solution version arn \
--min-provisioned-tps 1
```

Actualizar una campaña (AWS SDK)

Para implementar una nueva versión de la solución, cambiar el TPS mínimo aprovisionado de la campaña o cambiar la configuración de la campaña, debe actualizarla. Usa el siguiente código para actualizar una campaña con AWS SDK para Python (Boto3) o con el SDK for Java 2.x. Para obtener una lista completa de parámetros, consulte UpdateCampaign.

SDK for Python (Boto3)

Use el siguiente método `update_campaign` para implementar una nueva versión de la solución. Reemplace `campaign arn` por el nombre de recurso de Amazon (ARN) de la campaña que desea actualizar, reemplace `new solution version arn` por el ARN de la nueva versión de solución y, si lo desea, cambie `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

Use el siguiente método `updateCampaign` para actualizar una campaña y usar una nueva versión de la solución. Pase como parámetros un cliente de servicio Amazon Personalize, el nombre de recurso de Amazon (ARN) de la nueva versión de la solución y TPS mínimo aprovisionado.

```
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);
    }
}
```

Obtención de recomendaciones de Amazon Personalize

Después de crear un recomendador o una campaña, podrá recibir recomendaciones. En función de sus recursos, puede obtener recomendaciones en tiempo real o mediante un flujo de trabajo por lotes.

- Con los recursos personalizados, puede obtener recomendaciones en tiempo real o recomendaciones por lotes. Para obtener recomendaciones en tiempo real, debe crear una campaña personalizada antes de recibir recomendaciones. En el caso de recomendaciones por lotes, no es necesario crear una campaña.
- Con los generadores de recomendaciones de un grupo de conjuntos de datos de dominios, solo puede obtener recomendaciones en tiempo real.

En los temas siguientes se explica cómo y cuándo utilizar cada tipo de recomendación. Con las recomendaciones en tiempo real y por lotes podrá filtrar los resultados. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Temas

- Puntuaciones de recomendaciones
- Recomendaciones de elementos en tiempo real en Amazon Personalize
- Recomendaciones de acciones en tiempo real en Amazon Personalize
- Obtención de una clasificación personalizada (recursos personalizados)
- Aumento de la relevancia de las recomendaciones con metadatos contextuales
- Obtención de recomendaciones de elemento por lotes con recursos personalizados
- Obtención de segmentos de usuarios por lotes con recursos personalizados

Puntuaciones de recomendaciones

Con soluciones personalizadas creadas con las recetas User-Personalization-v 2, Usuer-Personalization, Personalized-Ranking-v 2, Personalized-Ranking y PERSONALIZED_ACTIONS, Amazon Personalize incluye una puntuación para cada elemento en las recomendaciones. Estas puntuaciones representan la certeza relativa que Amazon Personalize tiene respecto a qué elemento o acción va a seleccionar el usuario a continuación. Las puntuaciones más altas representan una mayor certeza.

- Para obtener información sobre las puntuaciones de 2 y la personalización por User-Personalization-v usuario, consulte. Cómo funciona la puntuación de las recomendaciones (recursos personalizados)
- Para obtener información sobre las puntuaciones de las recetas PERSONALIZED_ACTIONS, consulte Cómo funciona la puntuación de recomendaciones de acciones.
- Para obtener información sobre las puntuaciones de Personalized-Ranking-v 2 y las recomendaciones de clasificación personalizada, consulte. Cómo funciona la puntuación de clasificación personalizada

Para los trabajos de inferencia por lotes, las puntuaciones de los elementos se calculan tal y como se describe en Cómo funciona la puntuación de las recomendaciones (recursos personalizados) y Cómo funciona la puntuación de clasificación personalizada. Puede ver las puntuaciones en el archivo JSON de salida del trabajo de inferencia por lotes.

Recomendaciones de elementos en tiempo real en Amazon Personalize

Si el caso de uso o la receta genera recomendaciones de elementos, después de crear un recomendador o crear una campaña, podrá obtener recomendaciones de elementos relacionados o personalizados en tiempo real para los usuarios.

Si el caso de uso de dominio la receta proporciona personalización en tiempo real, como el caso de uso Mejores opciones o la receta User-Personalization-v2, Amazon Personalize actualiza las recomendaciones en función de la actividad más reciente del usuario a medida que registra las interacciones con el catálogo. Para obtener más información sobre el registro de eventos en tiempo real y la personalización, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.

Cuando obtengas recomendaciones de elementos en tiempo real, puede hacer lo siguiente:

- Si ha configurado la campaña de forma que devuelva metadatos para los elementos recomendados, puede especificar las columnas que se incluirán en la operación de la API GetRecommendations. También puede especificar las columnas al probar la campaña con la consola de Amazon Personalize. Para obtener ejemplos de código, consulte Obtención de metadatos de elemento con recomendaciones en tiempo real. Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Metadatos de elemento en

las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador de dominios en Amazon Personalize.

- Para algunos casos de uso y recetas, puede especificar una promoción en la solicitud de recomendación. Una promoción define reglas empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados. Para obtener más información, consulte Promoción de elementos en las recomendaciones en tiempo real.
- Puede filtrar los resultados en función de criterios personalizados. Por ejemplo, es posible que no desee recomendar productos que un usuario ya haya comprado o recomendar solo elementos para un grupo de edad concreto. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

> **Note**
>
> Si ha utilizado una receta personalizada PERSONALIZED_RANKING, consulte Obtención de una clasificación personalizada (recursos personalizados).

Temas

- Cómo funciona la puntuación de las recomendaciones (recursos personalizados)
- Motivos de recomendación con User-Personalization-v2
- Obtención de recomendaciones de elementos en tiempo real
- Obtención de metadatos de elemento con recomendaciones en tiempo real
- Promoción de elementos en las recomendaciones en tiempo real

Cómo funciona la puntuación de las recomendaciones (recursos personalizados)

Con las recetas User-Personalization-v2 y User-Personalization, Amazon Personalize genera puntuaciones para los elementos en función de los metadatos y los datos de interacción de un usuario. Estas puntuaciones representan la certeza relativa que Amazon Personalize tiene respecto a qué elemento va a seleccionar el usuario a continuación. Las puntuaciones más altas representan una mayor certeza.

> **Note**
>
> Amazon Personalize no muestra las puntuaciones de los generadores de recomendaciones de dominio o las recetas Similar-Items, SIMS o Popularity-Count. Para obtener información sobre las puntuaciones de las recomendaciones de Personalized-Ranking, consulte Cómo funciona la puntuación de clasificación personalizada.

Amazon Personalize genera puntuaciones de elementos de forma relativa entre ellos en una escala de 0 a 1 (ambos incluidos). Con User-Personalization-v2, Amazon Personalize genera puntuaciones para un subconjunto de los elementos. Con User-Personalization, Amazon Personalize puntúa todos los elementos del catálogo.

Si utiliza User-Personalization-v2 y aplica un filtro a la recomendaciones, en función del número de recomendaciones que elimina el filtro, Amazon Personalize podría agregar elementos de marcador de posición. Lo hace para alcanzar el valor de `numResults` correspondiente a la solicitud de recomendación. Estos elementos son populares, según el volumen los datos de interacciones, que cumplen los criterios de filtro. No tienen una puntuación de relevancia para el usuario.

Tanto para User-Personalization-v2 como para User-Personalization, el total de todas las puntuaciones es igual a 1. Por ejemplo, si recibe recomendaciones de películas para un usuario y hay tres películas que aparecen en los conjuntos de datos de elementos e interacciones, las puntuaciones podrán ser `0.6`, `0.3` y `0.1`. Del mismo modo, si tiene 10 000 películas en el inventario, las películas con mayor puntuación pueden tener puntuaciones muy bajas (la puntuación media sería `.001`), pero, dado que la puntuación es relativa, las recomendaciones siguen siendo válidas.

En términos matemáticos, las puntuaciones de cada par de usuario y elemento (u, i) se calculan de acuerdo con la siguiente fórmula, donde `exp` es la función exponencial, $\bar{w}_u$ y $w_i/_j$ son incrustaciones de usuarios y elementos respectivamente, y la letra griega sigma (Σ) representa el sumatorio de todos los elementos con puntuaciones:

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_j \exp(\bar{w}_u^\top w_j)}$$

Motivos de recomendación con User-Personalization-v2

Si utiliza User-Personalization-v2, los elementos que el modelo normalmente no recomendaría incluyen una lista `reason`. Estos motivos explican por qué el elemento se ha incluido en las recomendaciones. Algunos motivos posibles son los siguientes:

- Elemento promocionado: indica que el elemento se ha incluido como parte de una promoción que ha aplicado en la solicitud de recomendación.
- Exploración: indica que el elemento se ha incluido con la exploración. Con la exploración, las recomendaciones incluyen elementos con menos datos de interacciones o relevancia para el usuario. Para obtener más información sobre la exploración, consulte Exploración.
- Elemento popular: indica que el elemento se ha incluido como marcador de posición de elemento popular. Si utiliza un filtro, en función del número de recomendaciones que elimina el filtro, Amazon Personalize podría agregar marcadores de posición de elemento para `numResults` correspondiente a la solicitud de recomendación. Estos elementos son populares, según los datos de interacciones, que cumplen los criterios de filtro. No tienen una puntuación de relevancia para el usuario.

Obtención de recomendaciones de elementos en tiempo real

Puede obtener recomendaciones de elementos en tiempo real de un recomendador o una campaña personalizada de Amazon Personalize con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Temas

- Obtención de recomendaciones de elementos (consola)
- Obtención de recomendaciones de elementos (AWS CLI)
- Obtención de recomendaciones de elementos (SDK de AWS)

Obtención de recomendaciones de elementos (consola)

Para obtener recomendaciones con la consola de Amazon Personalize, proporcione la información de la solicitud en la página de detalles de un generador de recomendaciones (grupo de conjuntos de datos de dominio) o una campaña personalizada.

Cómo obtener recomendaciones

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Elija el grupo de conjuntos de datos que contenga la campaña o el recomendador que esté utilizando.
3. En el panel de navegación, elija Campañas o Generadores de recomendaciones.
4. Elija la campaña o el recomendador objetivo.
5. En el caso de una campaña, en Probar resultados de la campaña, introduzca los detalles de la solicitud de recomendación en función de la receta que haya utilizado. En el caso de un generador de recomendaciones, elija Probar recomendador e introduzca los detalles de su solicitud de recomendación en función de su caso de uso.

   Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.
6. Si lo desea, elija un filtro. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.
7. Si utiliza metadatos contextuales, proporcione datos para cada contexto. Para cada contexto, introduzca el campo de metadatos para la Clave. Para el Valor, introduzca los datos de contexto. Para obtener más información, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales.
8. Si ha habilitado los metadatos en las recomendaciones para su campaña o recomendador, en Columnas de conjuntos de datos de elementos, elija las columnas de metadatos que quiera incluir en los resultados de las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Metadatos de elemento en las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador de dominios en Amazon Personalize.
9. Si desea promocionar un subconjunto de elementos, si lo desea, complete los campos de Promoción. Para obtener más información consulte () Promoción de elementos en las recomendaciones en tiempo real.
10. Elija Obtener recomendaciones. Aparece una tabla con los 25 elementos más recomendados por el usuario. Si utiliza User-Personalization-v2, cada elemento recomendado incluye una lista

de los motivos por los que se ha incluido el elemento en las recomendaciones. Para obtener más información, consulte Motivos de recomendación con User-Personalization-v2.

Obtención de recomendaciones de elementos (AWS CLI)

Utilice el siguiente código para obtener recomendaciones de una campaña. Para obtener recomendaciones de un recomendador, reemplace el parámetro `campaign-arn` por `recommender-arn`.

Especifique el ID del usuario para el que desea obtener recomendaciones, y el nombre de recurso de Amazon (ARN) de su campaña o generador de recomendaciones. Se muestra una lista de los 10 elementos más recomendados para el usuario. Si utiliza User-Personalization-v2, cada elemento recomendado incluye una lista de los motivos por los que se ha incluido el elemento en las recomendaciones. Para obtener más información, consulte Motivos de recomendación con User-Personalization-v2.

Para cambiar el número de elementos recomendados, cambie el valor de `numResults`. El valor predeterminado es de 25 elementos. El máximo es de 500 elementos. Si ha utilizado una receta RELATED_ITEMS para entrenar la versión de la solución que respalda la campaña, sustituya el parámetro `user-id` por `item-id` y especifique el ID de elemento.

Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

Obtención de recomendaciones de elementos (SDK de AWS)

El siguiente código muestra cómo obtener recomendaciones de Amazon Personalize para un usuario a partir de una campaña con los SDK de AWS. Para obtener recomendaciones de un recomendador, reemplace el parámetro `campaignArn` por `recommenderArn`.

Especifique el ID del usuario para el que desea obtener recomendaciones, y el nombre de recurso de Amazon (ARN) de su campaña o generador de recomendaciones. Se muestra una lista de

los 10 elementos más recomendados para el usuario. Si utiliza User-Personalization-v2, cada elemento recomendado incluye una lista de los motivos por los que se ha incluido el elemento en las recomendaciones. Para obtener más información, consulte Motivos de recomendación con User-Personalization-v2.

Para cambiar el número de elementos recomendados, cambie el valor de `numResults`. El valor predeterminado es de 25 elementos. El máximo es de 500 elementos. Si ha utilizado una receta RELATED_ITEMS para entrenar la versión de la solución que respalda la campaña, sustituya el parámetro `userId` por `itemId` y especifique el ID de elemento.

Si ha habilitado los metadatos en las recomendaciones para su campaña o recomendador, puede especificar las columnas de metadatos del conjunto de datos de elementos para incluirlas en la respuesta. Para ver una muestra de código, consulte Inclusión de metadatos de elemento con recomendaciones (SDK de AWS). Para obtener más información acerca de cómo habilitar los metadatos, consulte Metadatos de elemento en las recomendaciones.

Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.

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();
```

Obtención de metadatos de elemento con recomendaciones en tiempo real

Si ha configurado la campaña o el recomendador de forma que devuelva metadatos para los elementos recomendados, puede especificar las columnas que se incluirán en la operación de la API GetRecommendations. También puede especificar las columnas al probar la campaña con la consola de Amazon Personalize.

Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Metadatos de elemento en las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador de dominios en Amazon Personalize.

En las siguientes muestras de código, se presenta cómo especificar las columnas de metadatos que se incluirán en la AWS CLI o los SDK de AWS. Para hacerlo con la consola de Amazon Personalize, especifique las columnas al probar la campaña. Para obtener más información, consulte Obtención de recomendaciones de elementos (consola).

Temas

- Inclusión de metadatos de elemento con recomendaciones (AWS CLI)
- Inclusión de metadatos de elemento con recomendaciones (SDK de AWS)

Inclusión de metadatos de elemento con recomendaciones (AWS CLI)

Si ha habilitado los metadatos en las recomendaciones para su campaña o recomendador, puede especificar las columnas de metadatos del conjunto de datos de elementos para incluirlas en

la respuesta. En el siguiente ejemplo de código se muestra cómo especificar las columnas de metadatos como parte de la solicitud de recomendaciones.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10 \
--metadata-columns "{\"ITEMS\": [\"\"columnNameA\"\",\"\"columnNameB\"\"]}"
```

Inclusión de metadatos de elemento con recomendaciones (SDK de AWS)

Si ha habilitado los metadatos en las recomendaciones para su campaña o recomendador, puede especificar las columnas de metadatos del conjunto de datos de elementos para incluirlas en la respuesta. En el siguiente ejemplo de código se muestra cómo especificar las columnas de metadatos como parte de la solicitud de recomendaciones.

```
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'])
```

Promoción de elementos en las recomendaciones en tiempo real

Con todos los casos de uso de dominio y algunas recetas personalizadas, puede especificar una promoción al recibir recomendaciones en tiempo real.

Una promoción define reglas empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados. Por ejemplo, es posible que tenga una aplicación de streaming y

desee promocionar sus propias series y películas, pero también recomendar títulos interesantes. Puede usar una promoción para especificar que un porcentaje determinado de los elementos recomendados deban provenir de la categoría interno. El resto de los elementos recomendados seguirán siendo recomendaciones pertinentes en función de su receta y de cualquier filtro de solicitud.

Para aplicar una promoción, debe especificar lo siguiente en su solicitud de recomendación:

- El porcentaje de elementos recomendados a los que se debe aplicar el filtro de promoción.
- Un filtro que especifica los criterios de promoción. Para obtener más información, consulte Filtros de promoción.

En la respuesta a la recomendación, los elementos promocionados se colocan aleatoriamente en relación con otros elementos recomendados, pero ordenados en relación con otros elementos promocionados. Según la receta, los elementos recomendados que no forman parte de una promoción se ordenan por relevancia para el usuario, popularidad o similitud. Si no hay suficientes elementos que cumplan los criterios de la promoción, el resultado incluirá tantos elementos promocionados como sea posible.

Puedes aplicar una promoción a las recomendaciones con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS los SDK.

Temas

- Casos de uso y recetas que admiten promociones
- Filtros de promoción
- Promoción de nuevos elementos
- Promoción de elementos (consola)
- Promocionar artículos (AWS CLI)
- Artículos de promoción (AWS SDK)

Casos de uso y recetas que admiten promociones

Todos los casos de uso admiten promociones. Las siguientes recetas personalizadas admiten promociones:

- Recetas User-Personalization-v2 y User-Personalization

- Recetas Similar-Items y SIMS
- Recetas Trending-Now y Popularity-Count

Filtros de promoción

Cuando aplica una promoción a una solicitud de recomendación, elige un filtro que especifica los criterios de la promoción. Puede usar un filtro existente o crear uno nuevo. Los filtros de las promociones se crean y administran igual que haría con otros filtros de Amazon Personalize. Para obtener más información acerca de cómo se crean y se administran filtros, consulte Filtrado de resultados.

La única diferencia entre un filtro de promoción y un filtro que elija fuera de la promoción (el filtro de solicitud) es la forma en que Amazon Personalize los aplica. Un filtro de promoción se aplica solo a los elementos en promoción, mientras que un filtro de solicitud se aplica solo al resto de los elementos recomendados. Si especifica un filtro de solicitud y un filtro de promoción y desea aplicar ambos filtros a los elementos en promoción, la expresión del filtro de promoción debe incluir ambas expresiones. La forma de combinar dos expresiones depende de los conjuntos de datos que utilice. Para obtener más información sobre las expresiones de filtro, sus reglas y cómo crearlas, consulte Expresiones de filtro.

Ejemplos de expresiones de filtro

La siguiente expresión incluye solo los elementos de la categoría “internos”. Puede usar esta expresión si desea promocionar su propio contenido en las recomendaciones.

```
INCLUDE ItemID WHERE Items.OWNER IN ("in-house")
```

La siguiente expresión incluye solo los elementos creados más recientes que la marca temporal que especifique. Puede utilizar esta expresión para promocionar nuevos elementos en las recomendaciones.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

La siguiente expresión muestra cómo puede aplicar un filtro de solicitudes a los elementos promocionados. Incluye solo las prendas de vestir disponibles como elementos promocionados. En este escenario, también se usaría `Items.AVAILABLE IN ("True")` en la expresión del filtro de solicitudes, de modo que todas las recomendaciones se refieran a los elementos que están disponibles.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") AND Items.AVAILABLE IN ("True")
```

Para obtener una lista más completa de ejemplos de filtro, consulte Ejemplos de expresiones de filtro.

Promoción de nuevos elementos

Si usa la Receta User-Personalization-v2, Amazon Personalize recomienda los elementos más relevantes a los usuarios en función de los datos y recomienda con mayor frecuencia los elementos existentes con datos de interacciones. Para asegurarse de que las recomendaciones incluyen algunos elementos nuevos, puede aplicar una promoción a las solicitudes de recomendación que incluya elementos en función de la marca temporal de creación.

Si aún no utiliza una promoción, la expresión de filtro puede promocionar elementos creados después de una fecha determinada:

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

Si ya utiliza una promoción, debe crear una expresión que encadene la promoción y las instrucciones de condición de nuevo elemento:

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") OR Items.CREATION_TIMESTAMP > $DATE
```

Promoción de elementos (consola)

Para promocionar determinados elementos de las recomendaciones con la consola Amazon Personalize, cree un filtro y, a continuación, proporcione los detalles de la promoción en la solicitud de recomendación. Para obtener información sobre los demás campos, consulte Obtención de recomendaciones de elementos (consola).

Para promocionar los elementos de las recomendaciones

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija el grupo de conjuntos de datos que contenga la campaña o el recomendador que esté utilizando.
3. Si aún no lo ha hecho, debe crear un filtro que especifique los criterios de promoción. Los filtros para las promociones se crean del mismo modo que los filtros de solicitudes. Para obtener más información acerca de cómo se crean y se administran filtros, consulte Filtrado de resultados.

4. En el panel de navegación, seleccione Recomendadores o Campañas.
5. Elija la campaña o el recomendador objetivo.
6. En el caso de las campañas, en Probar resultado de la campaña, introduzca los detalles de la solicitud de recomendación en función de la receta que haya utilizado. En el caso de los recomendadores, seleccione Probar recomendador e indique detalles de la solicitud de recomendación.
7. Si lo desea, elija un filtro para la solicitud. Este filtro se aplica únicamente a los elementos no promocionados. Para obtener más información acerca de cómo se crean y se administran filtros, consulte Filtrado de resultados.
8. Si utiliza metadatos contextuales, proporcione datos para cada contexto. Para cada contexto, introduzca el campo de metadatos para la Clave. Para el Valor, introduzca los datos de contexto. Para obtener más información, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales.
9. Para la promoción, especifique lo siguiente:

   - Porcentaje de elementos promocionados: indique el porcentaje de elementos recomendados a los que se aplicará la promoción.
   - Filtro: elija un filtro que especifique los criterios de promoción. Este filtro se aplica a los elementos promocionados en lugar de a cualquier filtro de solicitud que haya especificado en el paso 7.
   - Parámetro de filtro: si la promoción usa un filtro con parámetros de marcador de posición, indique el valor para establecer los criterios del filtro para cada parámetro. Para usar varios valores para un parámetro, separe cada valor con una coma.

10. Elija Obtener recomendaciones. Aparece una tabla con los 25 elementos más recomendados por el usuario. La columna Artículo promocionado indica si el elemento se incluyó debido a su promoción. Los elementos promocionados se colocan aleatoriamente en relación con otros elementos recomendados, pero ordenados en relación con otros elementos promocionados. Según el caso de uso o la receta, los elementos recomendados que no forman parte de una promoción se ordenan por relevancia para el usuario, popularidad o similitud. Si no hay suficientes elementos que cumplan los criterios de la promoción, el resultado incluirá tantos elementos promocionados como sea posible.

Promocionar artículos (AWS CLI)

En el siguiente código se muestra cómo promocionar los artículos de las recomendaciones con una campaña personalizada. AWS CLI Para promocionar elementos con un recomendador, sustituya el parámetro `campaign-arn` por `recommender-arn` y especifique el nombre de recurso de Amazon (ARN) del recomendador. Para los campos de promoción, especifique lo siguiente:

- name: asigne un nombre a la promoción. La respuesta a la recomendación utiliza el nombre para identificar los elementos promocionados.
- percent-promoted-items: el porcentaje de elementos recomendados a los que se aplicará la promoción. En este ejemplo, el 50 % de los elementos serán elementos promocionados.
- filterArn: especifique el nombre de recurso de Amazon (ARN) del filtro que define los criterios de promoción. Para obtener más información, consulte Filtros de promoción.
- nombres y valores de los parámetros: si la expresión de filtro tiene algún parámetro, proporcione los nombres de los parámetros (teniendo en cuenta que se distingue entre mayúsculas y minúsculas) y los valores. Por ejemplo, si la expresión de filtro tiene un parámetro `$GENRE`, indique GENRE como la clave y uno o varios géneros, como Comedia, como valor. Separe los valores con comas. Si usas el AWS CLI, para cada valor debes usar el `/` carácter para evitar tanto las comillas como el `/` carácter. El siguiente ejemplo de código muestra cómo aplicar formato a los valores.

El código muestra cómo utilizar un filtro de solicitud y un filtro de promoción. Un filtro de promoción se aplica solo a los elementos en promoción, mientras que un filtro de solicitud se aplica solo al resto de los elementos recomendados. Para obtener más información, consulte Filtros de promoción.

Para obtener más información sobre los campos adicionales, consulte Obtención de recomendaciones de elementos (SDK de AWS) y Obtención de una clasificación personalizada con metadatos contextuales.

```
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\\\"\"}
}]"
```

Aparece una lista de elementos recomendados. Los elementos promocionados se colocan aleatoriamente en relación con otros elementos recomendados, pero ordenados en relación con otros elementos promocionados. Según la receta, los elementos recomendados que no forman parte de una promoción se ordenan por relevancia para el usuario, popularidad o similitud. Si no hay suficientes elementos que cumplan los criterios de la promoción, el resultado incluirá tantos elementos promocionados como sea posible.

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

Artículos de promoción (AWS SDK)

En el código siguiente se muestra cómo promocionar los elementos de las recomendaciones con el SDK for Java 2.x y una campaña personalizada. AWS SDK para Python (Boto3) Para promocionar elementos con un recomendador, sustituya el parámetro `campaignArn` por `recommenderArn` y especifique el nombre de recurso de Amazon (ARN) del recomendador. Para los campos de promoción, especifique lo siguiente:

- name: especifique el nombre de la promoción. La respuesta a la recomendación incluye el nombre para identificar los elementos promocionados.

- porcentajePromotedItems: el porcentaje de artículos recomendados a los que aplicar la promoción.
- promotionFilterARN: el nombre de recurso de Amazon (ARN) del filtro que define los criterios de promoción. Para obtener más información, consulte Filtros de promoción.
- Nombres y valores de los parámetros: si la expresión de filtro tiene algún parámetro, proporcione los nombres de los parámetros (teniendo en cuenta que se distingue entre mayúsculas y minúsculas) y los valores para cada uno de ellos. Por ejemplo, si la expresión de filtro tiene un parámetro `$GENRE`, indique `"GENRE"` como la clave y uno o varios géneros, como `"\"Comedia"\"`, como valor. Separe los valores con comas. Por ejemplo, `"\"comedy\",\"drama\", \"horror"\"`.

El código siguiente muestra cómo utilizar un filtro de solicitud y un filtro de promoción. Un filtro de promoción se aplica solo a los elementos en promoción, mientras que un filtro de solicitud se aplica solo al resto de los elementos recomendados. Para obtener más información, consulte Filtros de promoción.

Para obtener más información sobre los campos adicionales, consulte Obtención de recomendaciones de elementos (SDK de AWS) y Obtención de una clasificación personalizada con metadatos contextuales.

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();
```

Aparece una lista de elementos recomendados. Los elementos promocionados se colocan aleatoriamente en relación con otros elementos recomendados, pero ordenados en relación con otros elementos promocionados. Según la receta, los elementos recomendados que no forman parte de una promoción se ordenan por relevancia para el usuario, popularidad o similitud. Si no hay suficientes elementos que cumplan los criterios de la promoción, el resultado incluirá tantos elementos promocionados como sea posible.

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

Recomendaciones de acciones en tiempo real en Amazon Personalize

Si utiliza una receta PERSONALIZED_ACTIONS, puede obtener recomendaciones de acciones de su campaña en tiempo real. Puede obtener recomendaciones de acciones con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Temas

- Cómo funciona la puntuación de recomendaciones de acciones
- Obtención de recomendaciones de acciones (consola)
- Obtención de recomendaciones de acciones (AWS CLI)

- Obtención de recomendaciones de acciones (SDK de AWS)

Cómo funciona la puntuación de recomendaciones de acciones

Con la receta Next-Best-Action, Amazon Personalize genera puntuaciones para las acciones en función de la probabilidad de que el usuario interactúe con la acción. Las puntuaciones pueden estar entre 0 y 1,0. Cuanto más se acerque a 1,0 este valor, mayores serán las probabilidades de que el usuario interactúe con la acción.

Si no ha importado datos de interacción de acciones, todas las acciones recomendadas tendrán una puntuación de 0,0. Si Amazon Personalize recomienda una acción como parte de la exploración, el elemento tendrá una puntuación de 0,0. Amazon Personalize usa la exploración para recomendar acciones sin datos de interacción de acciones. Para obtener más información acerca de la exploración, consulte Exploration (Exploración).

Obtención de recomendaciones de acciones (consola)

Para obtener recomendaciones de acciones con la consola de Amazon Personalize, proporcione la información de la solicitud en la página de detalles de su campaña personalizada.

Para obtener recomendaciones de acciones

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Elija el grupo de conjuntos de datos que contiene la campaña que está usando.
3. En el panel de navegación, en Recursos personalizados, elija Campañas.
4. Elija la campaña de destino.
5. En Resultados de la campaña de prueba, introduzca los detalles de su solicitud de recomendación.

   Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.
6. Si lo desea, elija un filtro. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

7. Elija Obtener recomendaciones. Aparece una tabla que contiene las 5 acciones principales recomendadas.

Obtención de recomendaciones de acciones (AWS CLI)

Utilice el siguiente código para obtener recomendaciones de acciones de una campaña. Especifique el ID del usuario para el que quiere obtener recomendaciones y el nombre de recurso de Amazon (ARN) de su campaña.

Para cambiar el número de acciones recomendadas, cambie el valor de `numResults`. El valor predeterminado es de 5 acciones. El máximo es de 100.

Para filtrar las recomendaciones de acciones por criterios personalizados, puede crear un filtro y aplicarlo a la operación `get-action-recommendations`. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.

```
aws personalize-runtime get-action-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

Obtención de recomendaciones de acciones (SDK de AWS)

El siguiente código muestra cómo obtener recomendaciones de Amazon Personalize para un usuario a partir de una campaña. Especifique el ID del usuario para el que quiere obtener recomendaciones y el nombre de recurso de Amazon (ARN) de su campaña.

Para cambiar el número de acciones recomendadas, cambie el valor de `numResults`. El valor predeterminado es de 5 acciones. El máximo es de 100.

Para filtrar las recomendaciones de acciones por criterios personalizados, puede crear un filtro y aplicarlo a la solicitud de API GetActionRecommendations. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Si ha registrado eventos para un usuario antes de que iniciara sesión (un usuario anónimo), puede obtener recomendaciones para ese usuario si proporciona el valor `sessionId` de esos eventos como si fuera su `userId`. Para obtener más información sobre el registro de eventos para usuarios anónimos, consulte Registro de eventos para usuarios anónimos.

```
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'])
```

Obtención de una clasificación personalizada (recursos personalizados)

Una clasificación personalizada es una lista de elementos recomendados que se vuelven a clasificar para un usuario específico. Para obtener clasificaciones personalizadas, llame a la operación de la API GetPersonalizedRanking u obtenga recomendaciones a partir de una campaña en la consola.

> **Note**
>
> La solución relacionada con la campaña se debe haber creado utilizando una receta del tipo PERSONALIZED_RANKING. Para obtener más información, consulte Elección de una receta.

Temas

- Cómo funciona la puntuación de clasificación personalizada
- Obtención de una clasificación personalizada (consola)
- Obtención de una clasificación personalizada (AWS CLI)

- Obtención de una clasificación personalizada (SDK de AWS)
- Cuaderno de ejemplo de Personalized-Ranking

Cómo funciona la puntuación de clasificación personalizada

Al igual que las puntuaciones que devuelve la operación `GetRecommendations` para las soluciones creadas con las recetas User-Personalization-v2 y User-Personalization, la suma de las puntuaciones de `GetPersonalizedRanking` es 1; sin embargo, solo los elementos de entrada reciben puntuaciones y las puntuaciones de recomendación suelen ser más altas. Si un objeto no estaba presente durante el último entrenamiento, recibe una puntuación de 0.

Matemáticamente, la función de puntuación para GetPersonalizedRanking es idéntica a `GetRecommendations`, excepto que solo considera los elementos de entrada. Esto significa que las puntuaciones más cercanas a 1 se vuelven más probables, ya que hay menos opciones entre las que dividir la puntuación:

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_{j \in \text{input}} \exp(\bar{w}_u^\top w_j)}$$

Obtención de una clasificación personalizada (consola)

Para obtener una clasificación personalizada para un usuario desde la consola de Amazon Personalize, elija la campaña que está utilizando y, a continuación, proporcione su ID de usuario, especifique la lista de elementos que desea clasificar para el usuario, elija un filtro y, si lo desea, proporcione cualquier dato de contexto.

Para obtener una clasificación personalizada para un usuario

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. Elija el grupo de conjuntos de datos que contiene la campaña que está usando.
3. En el panel de navegación, seleccione Campañas.
4. En la página Campañas, elija la campaña objetivo.
5. En Probar resultados de la campaña, introduzca el ID de usuario para el que quiera recibir recomendaciones.
6. En el ID de elementos, especifique la lista de elementos que se clasificarán para el usuario.

7. Si lo desea, elija un filtro. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.
8. Si ha habilitado los metadatos en las recomendaciones para su campaña, en Columnas de conjuntos de datos de elementos, elija las columnas de metadatos que quiera incluir en los resultados de las recomendaciones. Para obtener más información acerca de cómo habilitar los metadatos, consulte Metadatos de elemento en las recomendaciones.
9. Si su campaña utiliza metadatos contextuales (para ver los requisitos, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales), si lo desea, proporcione datos contextuales.

   Para cada contexto, para la Clave, especifique el campo de metadatos y, para el Valor, especifique los datos contextuales.
10. Elija Obtener clasificaciones de elementos personalizadas. Aparece una tabla que contiene los elementos clasificados en orden de interés previsto para el usuario.

Obtención de una clasificación personalizada (AWS CLI)

Los siguientes ejemplos de código muestran distintas variaciones de cómo obtener una clasificación personalizada con la AWS CLI.

Temas

- Obtención de una clasificación personalizada
- Incluir los metadatos de los elementos en una clasificación personalizada

Obtención de una clasificación personalizada

Utilice el siguiente comando `get-personalized-ranking` para obtener una clasificación personalizada con el AWS CLI. Especifique el nombre de recurso de Amazon (ARN) de su campaña, el ID de usuario, y proporcione una lista de los ID de elementos que se clasificarán para el usuario (cada uno separado por un espacio). Los elementos que se clasificarán deben estar en los datos que utilizó para preparar la versión de la solución. Se muestra una lista de recomendaciones clasificadas. Amazon Personalize considera el primer elemento de la lista que más interesa al usuario.

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
```

```
--input-list 3 4 10 8 12 7
```

Incluir los metadatos de los elementos en una clasificación personalizada

Si ha habilitado los metadatos en las recomendaciones para su campaña, puede especificar las columnas de metadatos del conjunto de datos de elementos para incluirlas en la respuesta. Para obtener más información acerca de cómo habilitar los metadatos, consulte Metadatos de elemento en las recomendaciones.

En el siguiente ejemplo de código se muestra cómo especificar las columnas de metadatos como parte de la solicitud de una clasificación personalizada.

```
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\"\"]}"
```

Obtención de una clasificación personalizada (SDK de AWS)

Los siguientes ejemplos de código muestran las diferentes variaciones de cómo obtener una clasificación personalizada con los SDK de AWS.

Temas

- Obtención de una clasificación personalizada
- Incluir los metadatos de los elementos en una clasificación personalizada
- Obtención de una clasificación personalizada con metadatos contextuales

Obtención de una clasificación personalizada

El siguiente código muestra cómo obtener una clasificación personalizada para un usuario. Especifique el ID del usuario y una lista de los ID de los elementos que se clasificarán para el usuario. Los ID de los elementos deben estar en los datos que utilizó para entrenar la versión de la solución. Se devuelve una lista de recomendaciones clasificadas. Amazon Personalize considera el primer elemento de la lista que más interesa al usuario.

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();
```

Incluir los metadatos de los elementos en una clasificación personalizada

Si ha habilitado los metadatos en las recomendaciones para su campaña, puede especificar las columnas de metadatos del conjunto de datos de elementos para incluirlas en la respuesta. Para

obtener más información acerca de cómo habilitar los metadatos, consulte Metadatos de elemento en las recomendaciones.

En el siguiente ejemplo de código se muestra cómo especificar las columnas de metadatos como parte de la solicitud de una clasificación personalizada.

```
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'])
```

Obtención de una clasificación personalizada con metadatos contextuales

Utilice el siguiente código para obtener una clasificación personalizada basada en metadatos contextuales. Para `context`, para cada par clave-valor, proporcione el campo de metadatos como la clave y los datos contextuales como el valor. En el siguiente código de ejemplo, la clave es `DEVICE` y el valor es `mobile phone`. Sustituya estos valores y el `Campaign ARN` y `User ID` por los suyos propios. Cambie también el valor de `inputList` por una lista de ID de elementos que se encuentren en los datos que utilizó para entrenar la solución. Amazon Personalize considera el primer elemento de la lista que más interesa al usuario.

```
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'])
```

Cuaderno de ejemplo de Personalized-Ranking

Para ver una muestra de cuaderno de Jupyter que describe el uso de la receta Personalized-Ranking, consulte el ejemplo de Personalize Ranking.

Aumento de la relevancia de las recomendaciones con metadatos contextuales

Para aumentar la relevancia de las recomendaciones, incluya metadatos contextuales para un usuario, como su tipo de dispositivo u hora del día, cuándo recibe recomendaciones de elementos u obtiene una clasificación personalizada.

Para usar metadatos contextuales, el esquema del conjunto de datos de interacciones entre elementos debe tener campos de metadatos para los datos contextuales. Por ejemplo, un campo DEVICE (consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize). Al considerar los campos que se van a usar como metadatos contextuales, seleccione los campos en los que los valores estén disponibles para datos históricos en el conjunto de datos de interacciones entre elementos y cuando reciba recomendaciones de elementos o obtenga una clasificación personalizada de los elementos.

Para los grupos de conjuntos de datos de dominio, los siguientes casos de uso de generador de recomendaciones pueden utilizar metadatos contextuales:

- Recomendaciones (dominio de ECOMMERCE)
- Mejores opciones (dominio de VIDEO_ON_DEMAND)

En el caso de los recursos personalizados, entre las recetas que utilizan metadatos contextuales se incluyen las siguientes:

- User-Personalization-v2 y User-Personalization
- Personalized-Ranking-v2 y Personalized-Ranking

Para obtener más información sobre la información contextual, consulte la siguiente entrada del blog de AWS Machine Learning: Aumentar la relevancia de sus recomendaciones de Amazon Personalize mediante el aprovechamiento de la información contextual.

Puede obtener recomendaciones con metadatos contextuales con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWSSDKs.

Obtener recomendaciones mediante metadatos contextuales (SDK de AWS Python)

Para aumentar la relevancia de las recomendaciones, incluya metadatos contextuales para un usuario, como su tipo de dispositivo u hora del día, cuándo recibe recomendaciones de elementos u obtiene una clasificación personalizada.

Utilice el código siguiente para obtener una recomendación basada en metadatos contextuales. Para `context`, para cada par clave-valor, proporcione el campo de metadatos como la clave y los datos contextuales como el valor. En el siguiente código de ejemplo, la clave es `DEVICE` y el valor es `mobile phone`. Sustituya estos valores y el `Campaign ARN` y `User ID` por los suyos propios. Si ha creado un generador de recomendación, sustituya `campaignArn` por `recommenderArn`. Se muestra una lista de elementos recomendados para el usuario.

```
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'])
```

Obtención de recomendaciones de elemento por lotes con recursos personalizados

Con recursos personalizados, puede obtener recomendaciones de elementos con un flujo por lotes asíncrono. Por ejemplo, puede obtener recomendaciones de productos para todos los usuarios de una lista de correo electrónico o similitudes entre elementos de todo un inventario.

Para obtener recomendaciones por lotes para elementos, puede utilizar un trabajo de inferencia por lotes. Un trabajo de inferencia por lotes es una herramienta que importa sus datos de entrada del lote desde un bucket de Amazon S3, utiliza la versión de su solución personalizada para generar recomendaciones de elementos y, a continuación, exporta las recomendaciones de elementos a un bucket de Amazon S3. Según la receta, sus datos de entrada son una lista de usuarios o elementos, o una lista de usuarios, cada uno de ellos con un conjunto de elementos.

Si su solución usa la receta de elementos similares y tiene un conjunto de datos de elementos con datos textuales y datos del título de los elementos, puede generar recomendaciones por lotes con temas para cada grupo de elementos. Para obtener más información, consulte Recomendaciones por lotes con temas del Generador de contenidos.

Después de crear una versión de solución personalizada, la forma en que los nuevos datos influyen en las recomendaciones de elemento depende del tipo, el método de importación y la receta personalizada que utilice. Para obtener información sobre cómo los nuevos datos influyen en las recomendaciones, consulte Cómo influyen los datos nuevos en las recomendaciones por lotes (recursos personalizados).

Temas

- Flujo de trabajo por lotes
- Directrices y requisitos
- Puntuación de flujos de trabajo por lotes
- Recomendaciones por lotes con temas del Generador de contenidos
- Preparación de los datos de entrada para las recomendaciones por lotes
- Creación de un trabajo de inferencia por lotes
- Ejemplos de salida de trabajo de inferencia por lotes

Flujo de trabajo por lotes

El flujo de trabajo por lotes es el siguiente:

1. Prepare y suba sus datos de entrada en formato JSON en un bucket de Amazon S3. El formato de sus datos de entrada depende de la receta que utilice. Consulte Preparación de los datos de entrada para las recomendaciones por lotes.
2. Cree una ubicación independiente para sus datos de salida, ya sea una carpeta o un bucket de Amazon S3 diferente.
3. Cree un trabajo de inferencia por lotes Consulte Creación de un trabajo de inferencia por lotes.
4. Cuando se complete la inferencia por lotes, recupere las recomendaciones de elementos de su ubicación de salida en Amazon S3.

Directrices y requisitos

Las siguientes son directrices y requisitos para obtener recomendaciones por lotes:

- Su rol de servicio de IAM de Amazon Personalize debe tener permiso para leer y añadir archivos a sus buckets de Amazon S3. Para obtener información acerca de cómo conceder permisos, consulte Política de roles de servicio para flujos de trabajo por lotes. Para obtener más información acerca de los permisos de buckets, consulte Ejemplos de políticas de usuario en la Guía para desarrolladores de Amazon Simple Storage Service. Si usa AWS Key Management Service (AWS KMS) para cifrado, debe conceder permiso a Amazon Personalize y a su rol de servicio de IAM de Amazon Personalize para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
- Para crear un trabajo de inferencia por lotes, debe crear una solución y una versión de la solución personalizadas. Sin embargo, no es necesario crear una campaña de Amazon Personalize. Si creó un grupo de conjuntos de datos de dominio, aún puede crear recursos personalizados.
- Para generar temas con recomendaciones, debe usar la receta Similar-Items. Además, debe tener un conjunto de datos de elementos con datos textuales y datos del título de los elementos. Para obtener más información sobre recomendaciones de temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.
- Sus datos de entrada deben tener el formato que se describe en Preparación de los datos de entrada para los segmentos de usuarios.
- No puede obtener recomendaciones por lotes con las recetas Trending-Now o Next-Best-Action.

- Si utiliza un filtro con parámetros de marcadores de posición, debe incluir los valores de los parámetros en los datos de entrada de un objeto `filterValues`. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.
- Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente) que para sus datos de entrada.
- Las recomendaciones por lotes pueden no ser exactamente las mismas que las recomendaciones en tiempo real. Esto se debe a que los trabajos de inferencia por lotes tardan más en completarse y solo tienen en cuenta los datos disponibles 15 minutos antes del inicio del trabajo.

Puntuación de flujos de trabajo por lotes

Las recomendaciones por lotes incluyen las siguientes puntuaciones:

- Con las recetas User-Personalization y Personalized-Ranking, Amazon Personalize calcula las puntuaciones de recomendaciones de trabajos de inferencia por lotes tal y como se describe en Cómo funciona la puntuación de las recomendaciones (recursos personalizados) y Cómo funciona la puntuación de clasificación personalizada. Puede ver las puntuaciones en el archivo JSON de salida del trabajo de inferencia por lotes.
- Con la receta Similar-Items, si recibe recomendaciones por lotes de temas, Amazon Personalize clasifica cada conjunto de elementos relacionados en función de la relevancia del tema para cada elemento. Cada artículo incluye una puntuación de 0 a 1. Cuanto más alta sea la puntuación, más relacionado estará el elemento con el tema. Para obtener más información sobre las recomendaciones con temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

Recomendaciones por lotes con temas del Generador de contenidos

> **Important**
>
> Al obtener recomendaciones por lotes con temas, se incurre en costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Si usa la Receta Similar-Items, el Generador de contenidos de Amazon Personalize puede agregar temas descriptivos a las recomendaciones por lotes. Generador de contenidos es una capacidad de inteligencia artificial generativa (IA generativa) administrada por Amazon Personalize.

Cuando recibe recomendaciones por lotes con temas, el Generador de contenidos de Amazon Personalize agrega un tema descriptivo para cada conjunto de elementos similares. El tema se basa en la descripción del elemento y los datos del nombre del elemento en su conjunto de datos de elementos. Amazon Personalize incluye los temas en la salida del trabajo de inferencia por lotes. Puede usar los temas para hacer más convincente el texto de su solicitud o de sus mensajes de marketing.

Por ejemplo, si recibe recomendaciones de elementos relacionados con comida para desayuno, Amazon Personalize puede generar un tema como Levántese con energía o Fundamental por la mañana. Puede usar el tema para reemplazar un título genérico de carrusel, como Con frecuencia se compran juntos. Otra opción es incorporar el tema a un correo electrónico promocional o una campaña de marketing para crear nuevas opciones de menú.

AWS no supervisa los temas de Content Generator. Para confirmar la calidad del tema, puede usar las puntuaciones generadas para cada elemento recomendado. Para obtener más información, consulte Clasificación y puntuación de las recomendaciones por lotes con temas.

Temas

- Regiones admitidas
- Directrices y requisitos
- Clasificación y puntuación de las recomendaciones por lotes con temas
- Generación de recomendaciones por lotes con temas

Regiones admitidas

El generador de contenido Amazon Personalize solo está disponible en las siguientes AWS regiones:

- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Oregón)
- Asia-Pacífico (Tokio)

Directrices y requisitos

A continuación se muestran las directrices y los requisitos para generar recomendaciones con temas:

- El archivo de entrada puede tener hasta 100 elementos. Para obtener información sobre los datos de entrada para las recomendaciones por lotes, consulte Preparación de los datos de entrada para las recomendaciones por lotes.
- Su solución debe usar la Receta Similar-Items.
- Debe tener un conjunto de datos de elementos con los siguientes datos. Estos datos pueden ayudar a generar temas más relevantes.
  - Debe tener un campo textual, como el campo DESCRIPTION. Para obtener más información sobre los datos textuales, consulte Metadatos de texto no estructurado.
  - Debe tener una columna de cadena con datos del nombre del elemento, como un campo TITLE.

  Si su conjunto de datos de elementos no tiene estos datos, puede agregarlos. Para obtener información sobre la actualización de los datos existentes, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Clasificación y puntuación de las recomendaciones por lotes con temas

Cuando recibe recomendaciones por lotes con temas, Amazon Personalize clasifica cada conjunto de elementos en función de la relevancia del tema para cada elemento. Cada elemento incluye una puntuación en un rango aproximado entre -0,1 y 0,6. Cuanto más alta sea la puntuación, más relacionado estará el elemento con el tema. Puede usar las puntuaciones a fin de establecer un umbral para mostrar solo los elementos estrechamente relacionados con el tema.

Por ejemplo, Amazon Personalize puede devolver el tema `For your sweet tooth` y artículos relacionados, con posibles puntuaciones de: caramelo (puntuación 0,19884521), chocolate (puntuación 0,17664525), manzana (puntuación 0,08994528), polo (puntuación 0,14294521), batata (puntuación 0,07794527) y zanahoria (puntuación 0,04994523). En su solicitud, puede agregar una regla para incluir solo los elementos con una puntuación igual a `.10` o superior, con la eliminación de frutas y verduras.

En el ejemplo siguiente se muestra el formato de salida de un trabajo de inferencia por lotes que genera recomendaciones sobre películas con temas.

```
{"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,-
...
```

Generación de recomendaciones por lotes con temas

Para generar recomendaciones por lotes con temas, complete el flujo de trabajo por lotes tal y como se describe en Flujo de trabajo por lotes. Prepare los datos de entrada de la misma forma que lo haría para una receta `RELATED_ITEMS`. Para ver un ejemplo, consulta Recetas RELATED_ITEMS.

Al crear el trabajo de inferencia por lotes, habilita la generación de temas y especifica la columna del título de elemento de su conjunto de datos de elementos.

- Para obtener información sobre el uso de la consola de Amazon Personalize para crear un trabajo de inferencia por lotes que genere temas, consulte Creación de un trabajo de inferencia por lotes.
- Para ver un ejemplo de código que muestra cómo utilizar el AWS SDK para Python (Boto3) para crear un trabajo de inferencia por lotes que genere temas, consulteCreación de un trabajo de inferencia por lotes que genere temas.

Preparación de los datos de entrada para las recomendaciones por lotes

Un trabajo de inferencia por lotes importa los datos JSON de entrada del lote desde un bucket de Amazon S3, utiliza su versión de solución personalizada para generar recomendaciones y, a continuación, exporta las recomendaciones a un bucket de Amazon S3. Para poder obtener recomendaciones por lotes, debe preparar y subir su archivo JSON a un bucket de Amazon S3. Le recomendamos que cree una carpeta de salida en su bucket de Amazon S3 o utilice un bucket de Amazon S3 de salida independiente. A continuación, puede ejecutar varios trabajos de inferencia por lotes utilizando la misma ubicación de datos de entrada.

Si utiliza un filtro con parámetros de marcadores de posición, por ejemplo, `$GENRE`, debe proporcionar los valores de los parámetros de un objeto `filterValues` en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.

Para preparar e importar datos

1. Aplique formato a los datos de entrada del lote en función de la receta. No puede obtener recomendaciones por lotes con la receta Trending-Now.

- En el caso de las recetas USER_PERSONALIZATION y Popularity-Count, sus datos de entrada son un archivo JSON con una lista de userIds
- Para las recetas RELATED_ITEMS, los datos de entrada son una lista de itemIds
- Para recetas PERSONALIZED_RANKING, sus datos de entrada son una lista de userIds, cada uno de ellos emparejado con una colección de itemIds

Separe cada fila con una línea nueva. Para ver ejemplos de datos de entrada, consulte Ejemplos de JSON de entrada y salida de trabajos de inferencia por lotes.

2. Suba el JSON de entrada a una carpeta de entrada de su bucket de Amazon S3. Para obtener más información, consulte Subida de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.
3. Cree una ubicación independiente para sus datos de salida, ya sea una carpeta o un bucket de Amazon S3 diferente. Al crear una ubicación independiente para el JSON de salida, puede ejecutar varios trabajos de inferencia por lotes con la misma ubicación de datos de entrada.
4. Cree un trabajo de inferencia por lotes. Amazon Personalize envía las recomendaciones de la versión de su solución a la ubicación de sus datos de salida.

Ejemplos de JSON de entrada y salida de trabajos de inferencia por lotes

Cómo aplica formato a los datos de entrada de la receta que utiliza. Si utiliza un filtro con parámetros de marcadores de posición, por ejemplo, `$GENRE`, debe proporcionar los valores de los parámetros de un objeto `filterValues` en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.

En las siguientes secciones se enumeran ejemplos de entradas y salidas JSON con el formato correcto para trabajos de inferencia por lotes. No puede obtener recomendaciones por lotes con la receta Trending-Now.

Temas

- Recetas de USER_PERSONALIZATION
- Recetas POPULAR_ITEMS (solo Popularity-Count)
- Recetas PERSONALIZED_RANKING
- Recetas RELATED_ITEMS

Recetas de USER_PERSONALIZATION

A continuación, se muestran ejemplos de entrada y salida JSON con formato correcto para las recetas USER_PERSONALIZATION. Si utiliza User-Personalization-v2, cada elemento recomendado incluye una lista de los motivos por los que se ha incluido el elemento en las recomendaciones. Esta lista puede estar vacía. Para obtener más información sobre los posibles motivos, consulte Motivos de recomendación con User-Personalization-v2.

Input

Separe cada `userId` con una nueva línea de la siguiente manera.

```
{"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.
...
```

Recetas POPULAR_ITEMS (solo Popularity-Count)

A continuación, se muestran ejemplos de entrada y salida JSON con formato correcto para la receta Popularity-Count. No puede obtener recomendaciones por lotes con la receta Trending-Now.

Input

Separe cada `userId` con una nueva línea de la siguiente manera.

```
{"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"]}}
...
```

Recetas PERSONALIZED_RANKING

A continuación, se muestran ejemplos de entrada y salida JSON con formato correcto para recetas PERSONALIZED_RANKING.

Input

Separe cada `userId` y la lista de `itemIds` que se clasificarán con una nueva línea de la siguiente manera.

```
{"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]}}
...
```

Recetas RELATED_ITEMS

A continuación, se muestran ejemplos de entrada y salida JSON con formato correcto para recetas RELATED_ITEMS.

Input

Separe cada `itemId` con una nueva línea de la siguiente manera.

```
{"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"]}}
...
```

A continuación, se muestran ejemplos de entrada y salida JSON con formato correcto para la receta Similar-Items con temas.

Input

Separe cada `itemId` con una nueva línea de la siguiente manera.

```
{"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
...
```

Creación de un trabajo de inferencia por lotes

Cree un trabajo de inferencia de lotes para obtener recomendaciones de elementos por lotes para los usuarios en función de los datos de entrada de Amazon S3. Los datos de entrada pueden ser una lista de usuarios o elementos (o ambos) en formato JSON. Puede crear un trabajo de inferencia por lotes con la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) o AWS SDKs.

Cuando crea un trabajo de inferencia por lotes, especifica las rutas de Amazon S3 en sus ubicaciones de entrada y salida. Amazon S3 se basa en prefijos. Si proporciona un prefijo para la ubicación de los datos de entrada, Amazon Personalize utilizará todos los archivos que coincidan con ese prefijo como datos de entrada. Por ejemplo, si proporciona `s3://amzn-s3-demo-bucket/folderName` y su bucket también tiene una carpeta con una ruta de `s3://amzn-s3-demo-bucket/folderName_test`, Amazon Personalize utiliza todos los archivos de ambas carpetas como datos de entrada. Para usar solo los archivos de una carpeta específica como datos de entrada, finalice la ruta de Amazon S3 con un delimitador de prefijo, como/: `s3://amzn-s3-demo-bucket/folderName/` Para obtener más información sobre cómo Amazon S3 organiza los objetos, consulte Organizar, enumerar y trabajar con sus objetos.

Para obtener más información sobre el flujo de trabajo por lotes en Amazon Personalize, incluidos los requisitos de permisos, la puntuación de las recomendaciones y la preparación e importación de los datos de entrada, consulte Obtención de recomendaciones de elemento por lotes con recursos personalizados.

Temas

- Creación de un trabajo de inferencia por lotes (consola)
- Creación de un trabajo de inferencia por lotes (AWS CLI)
- Creación de un trabajo de inferencia por lotes (AWS SDKs)

Creación de un trabajo de inferencia por lotes (consola)

Una vez que haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de inferencia por lotes. En este procedimiento, se da por hecho que ya ha creado una solución y una versión de la solución (modelo entrenado).

Para crear un trabajo de inferencia por lotes (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.

2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En el panel de navegación, en Recursos personalizados, elija Trabajos de inferencia por lotes.
4. Elija Crear trabajo de inferencia por lotes.
5. Elija el tipo de trabajo de inferencia por lotes.

   - Para generar recomendaciones de elementos sin temas, elija Recomendaciones de elementos.
   - Si usa la receta Similar-Items y quiere agregar temas descriptivos a grupos de elementos similares, elija Recomendaciones de temas con el Generador de contenidos. Para generar temas, debe tener un conjunto de datos de elementos con datos de los nombres de elementos y datos textuales. Para obtener más información, consulte Recomendaciones por lotes con temas del Generador de contenidos.
6. En Detalles del trabajo de inferencia por lotes, en Nombre del trabajo de inferencia por lotes, especifique un nombre para el trabajo de inferencia por lotes.
7. En Solución, elija la solución y, a continuación, elija el ID de versión de la solución que desee utilizar para generar las recomendaciones.
8. En Número de resultados, si lo desea, especifique el número de recomendaciones para cada línea de datos de entrada. El valor predeterminado es 25.
9. Si el trabajo por lotes genera recomendaciones con temas, en Detalles de las recomendaciones de temas, elija la columna que contiene los nombres o los títulos de los elementos de su conjunto de datos de elementos. Estos datos pueden ayudar a generar temas más relevantes. Para obtener más información, consulte Recomendaciones por lotes con temas del Generador de contenidos.
10. En Origen de la entrada, especifique la ruta de Amazon S3 al archivo de entrada.

    Utilice la siguiente sintaxis: **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`**

    Los datos de entrada deben tener el formato correcto para la receta que utiliza su solución. Para ver ejemplos de datos de entrada, consulteEjemplos de JSON de entrada y salida de trabajos de inferencia por lotes.
11. En el caso de la clave de descifrado, si utiliza su propia AWS KMS clave para el cifrado de buckets, especifique el nombre de recurso de Amazon (ARN) de la clave. Amazon Personalize debe tener permiso para usar su clave. Para obtener información sobre cómo conceder permisos, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

12. En Destino de salida, especifique la ruta a su ubicación de salida. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente).

    Utilice la siguiente sintaxis: **`s3://amzn-s3-demo-bucket/<output folder name>/`**

13. Para la clave de cifrado, si utiliza su propia AWS KMS clave para el cifrado, especifique el ARN de la clave. Amazon Personalize debe tener permiso para usar su clave. Para obtener información sobre cómo conceder permisos, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
14. Para el Rol de servicio de IAM, elija el rol de servicio de IAM que creó para Amazon Personalize durante la configuración. Este rol debe tener acceso de lectura y escritura a sus buckets de entrada y salida de Amazon S3, respectivamente.
15. En Filtros, puede elegir un filtro para aplicarlo a las recomendaciones por lotes. Si el filtro utiliza parámetros de marcador de posición, asegúrese de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.
16. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
17. Elija Crear trabajo de inferencia por lotes. Se inicia la creación del trabajo de inferencia por lotes y aparece la página Trabajos de inferencia por lotes con la sección Detalles del trabajo de inferencia por lotes visible.

    Cuando el estado del trabajo de inferencia por lotes cambia a Activo, puede recuperar la salida del trabajo en el bucket de Amazon S3 de salida designado. El nombre del archivo de salida tendrá el formato *`input-name`*`.out`.

Creación de un trabajo de inferencia por lotes (AWS CLI)

Cuando haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de inferencia por lotes con la operación CreateBatchInferenceJob.

Temas

- Creación de un trabajo de inferencia por lotes
- Creación de un trabajo de inferencia por lotes que genere temas

Creación de un trabajo de inferencia por lotes

Puede usar el comando `create-batch-inference-job` para crear un trabajo de inferencia por lotes. Especifique un nombre de trabajo, sustituya `Solution version ARN` por el Nombre de recurso de Amazon (ARN) de la versión de su solución y sustituya el `IAM service role ARN` por el ARN del rol de servicio de IAM que creó para Amazon Personalize durante la configuración. Este rol debe tener acceso de lectura y escritura a sus buckets de entrada y salida de Amazon S3, respectivamente. Si lo desea, proporcione un ARN de filtro para filtrar las recomendaciones. Si el filtro utiliza parámetros de marcador de posición, asegúrese de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados).

Sustituya `S3 input path` y `S3 output path` por la ruta de Amazon S3 al archivo de entrada y a las ubicaciones de salida. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice la siguiente sintaxis para las ubicaciones de entrada y salida: **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** y **`s3://amzn-s3-demo-bucket/<output folder name>/`**.

El ejemplo incluye hiperparámetros de `itemExplorationConfig` opcionales específicos de la receta User-Personalization: `explorationWeight` y `explorationItemAgeCutOff`. Si lo desea, incluya los valores `explorationWeight` y `explorationItemAgeCutOff` para configurar la exploración. Para obtener más información, consulte Receta User-Personalization.

```
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\"}}"
```

Creación de un trabajo de inferencia por lotes que genere temas

Si quiere generar temas para elementos similares, debe usar la receta Similar-Items y el conjunto de datos de elementos debe tener un campo textual y una columna de datos de nombre de elemento. Para obtener más información sobre las recomendaciones con temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

El siguiente código crea un trabajo de inferencia por lotes que genera recomendaciones con temas. Deje el valor `batch-inference-job-mode` establecido en `THEME_GENERATION`. Sustituya `COLUMN_NAME` por el nombre de la columna que almacena los datos de nombre de elemento.

```
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\"}}"
```

Creación de un trabajo de inferencia por lotes (AWS SDKs)

Cuando haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de inferencia por lotes con la operación CreateBatchInferenceJob.

Temas

- Creación de un trabajo de inferencia por lotes
- Creación de un trabajo de inferencia por lotes que genere temas

Creación de un trabajo de inferencia por lotes

Puede usar el siguiente código para crear un trabajo de inferencia por lotes. Especifique un nombre de trabajo, el nombre de recurso de Amazon (ARN) de la versión de la solución y el ARN del rol de servicio de IAM que creó para Amazon Personalize durante la configuración. Este rol debe tener acceso de lectura y escritura a los buckets de entrada y salida de Amazon S3.

Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice la siguiente sintaxis para las ubicaciones de entrada y salida: **`s3:/amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** y **`s3://amzn-s3-demo-bucket/<output folder name>/`**.

Para `numResults`, especifica el número de elementos que desea que Amazon Personalize prediga para cada línea de datos de entrada. Si lo desea, proporcione un ARN de filtro para filtrar las recomendaciones. Si el filtro utiliza parámetros de marcador de posición, asegúrese

de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados).

SDK for Python (Boto3)

El ejemplo incluye hiperparámetros de `itemExplorationConfig` opcionales específicos de la receta User-Personalization: `explorationWeight` y `explorationItemAgeCutOff`. Si lo desea, incluya los valores `explorationWeight` y `explorationItemAgeCutOff` para configurar la exploración. Para obtener más información, consulte Receta User-Personalization.

```
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

El ejemplo incluye hiperparámetros de `itemExplorationConfig` opcionales específicos de la receta User-Personalization: `explorationWeight` y `explorationItemAgeCutOff`. Si lo desea, incluya los valores `explorationWeight` y `explorationItemAgeCutOff` para configurar la exploración. Para obtener más información, consulte Receta User-Personalization.

```
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();
```

El procesamiento del trabajo por lotes puede tardar un tiempo en completarse. Puede comprobar el estado de un trabajo llamando a DescribeBatchInferenceJob y pasando un `batchRecommendationsJobArn` como el parámetro de entrada. También puede enumerar todos los trabajos de inferencia por lotes de Amazon Personalize de su AWS entorno llamando al teléfonoListBatchInferenceJobs.

Creación de un trabajo de inferencia por lotes que genere temas

Si quiere generar temas para elementos similares, debe usar la receta Similar-Items y el conjunto de datos de elementos debe tener un campo textual y una columna de datos de nombre de elemento. Para obtener más información sobre las recomendaciones con temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

El siguiente código crea un trabajo de inferencia por lotes que genera recomendaciones con temas. Deje el valor `batchInferenceJobMode` establecido en `"THEME_GENERATION"`. Sustituya `COLUMNN_NAME` por el nombre de la columna que almacena los datos de nombre de elemento.

```
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>/"}}
)
```

Ejemplos de salida de trabajo de inferencia por lotes

Al crear un trabajo de inferencia por lotes, el trabajo importa los datos de entrada del lote desde un bucket de Amazon S3, utiliza la versión de la solución para generar recomendaciones de elementos y exporta las recomendaciones a un bucket de Amazon S3 en formato JSON.

En las siguientes secciones se enumeran ejemplos de archivos de salida para trabajos de inferencia por lotes por tipo de receta. No puede obtener recomendaciones por lotes con las recetas Trending-Now o Next-Best-Action.

Temas

- Recetas de USER_PERSONALIZATION
- Recetas de POPULAR_ITEMS
- Recetas PERSONALIZED_RANKING
- Recetas de RELATED_ITEMS

Recetas de USER_PERSONALIZATION

A continuación se muestra un ejemplo del archivo JSON de salida para una receta de 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
...
```

Recetas de POPULAR_ITEMS

El siguiente ejemplo muestra el formato del archivo JSON de salida para la receta Popularity-Count. No puede obtener recomendaciones por lotes con la receta 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"]}}
...
```

Recetas PERSONALIZED_RANKING

En el ejemplo siguiente, se muestra el formato del archivo JSON de salida para una receta de 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]}}
...
```

Recetas de RELATED_ITEMS

En el ejemplo siguiente, se muestra el formato del archivo JSON de salida para una receta de 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"]}}
...
```

En el ejemplo siguiente, se muestra el formato del archivo JSON de salida para la receta Similar-Items con temas. Para obtener más información sobre las recomendaciones con temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

```
{"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,-
...
```


Obtención de segmentos de usuarios por lotes con recursos personalizados

Para obtener segmentos de usuarios, puede utilizar un trabajo de segmento por lotes. Un trabajo de segmento por lotes es una herramienta que importa los datos de entrada del lote desde un bucket de Amazon S3 y utiliza la versión de la solución entrenada con una receta USER_SEGMENTATION para generar segmentos de usuarios para cada fila de los datos de entrada.

Según la receta, los datos de entrada son una lista de elementos o atributos de metadatos de los elementos en formato JSON. Para los atributos de elementos, los datos de entrada pueden incluir expresiones para crear segmentos de usuarios basados en varios atributos de metadatos. Un trabajo de segmento por lotes exporta segmentos de usuarios a un bucket de salida de Amazon S3. Cada segmento de usuario se ordena en orden descendente en función de la probabilidad de que cada usuario interactúe con el elemento de los datos de entrada.

Al generar segmentos de usuarios, Amazon Personalize tiene en cuenta los datos de los conjuntos de datos procedentes de importaciones masivas e individuales:

- Para datos masivos, Amazon Personalize genera segmentos utilizando únicamente los datos masivos presentes en el último entrenamiento sobre la versión completa de la solución. Además, solo utiliza datos masivos que haya importado con el modo de importación COMPLETO (en sustitución de los datos existentes).
- Para los datos de las operaciones de importación de datos individuales, Amazon Personalize genera segmentos de usuarios con los datos presentes en el último entrenamiento sobre la versión completa de la solución. Para que los registros más recientes afecten a los segmentos de usuarios, cree una nueva versión de la solución y, a continuación, cree un trabajo de segmento por lotes.

La generación de segmentos de usuarios funciona de la siguiente manera:

1. Prepare y suba sus datos de entrada en formato JSON en un bucket de Amazon S3. El formato de los datos de entrada depende de la receta que utilice y del trabajo que esté creando. Consulte Preparación de los datos de entrada para los segmentos de usuarios.
2. Cree una ubicación independiente para sus datos de salida, ya sea una carpeta diferente o un bucket de Amazon S3 distinto.
3. Cree un trabajo de segmento por lotes. Consulte Obtención de segmentos de usuario con un trabajo de segmentos por lotes.
4. Cuando se complete el trabajo de segmentación por lotes, recupere los segmentos de usuario de su ubicación de salida en Amazon S3.

Temas

- Directrices y requisitos para obtener segmentos de usuarios
- Preparación de los datos de entrada para los segmentos de usuarios
- Obtención de segmentos de usuario con un trabajo de segmentos por lotes
- Ejemplos de formato de resultados de trabajos de segmento por lotes

Directrices y requisitos para obtener segmentos de usuarios

Las siguientes son directrices y requisitos para obtener segmentos por lotes:

- Debe usar una receta USER_SEGMENTATION.
- Su rol de servicio de IAM de Amazon Personalize necesita permiso para leer y añadir archivos a sus buckets de Amazon S3. Para obtener información acerca de cómo conceder permisos, consulte Política de roles de servicio para flujos de trabajo por lotes. Para obtener más información acerca de los permisos de buckets, consulte Ejemplos de políticas de usuario en la Guía para desarrolladores de Amazon Simple Storage Service.

  Si usa AWS Key Management Service (AWS KMS) para cifrado, debe conceder permiso a Amazon Personalize y a su rol de servicio de IAM de Amazon Personalize para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
- Para crear un trabajo de inferencia por lotes, debe crear una solución y una versión de la solución personalizadas. Sin embargo, no es necesario crear una campaña de Amazon Personalize. Si creó un grupo de conjuntos de datos de dominio, aún puede crear recursos personalizados.

- Sus datos de entrada deben tener el formato que se describe en Preparación de los datos de entrada para los segmentos de usuarios.
- Si utiliza la receta Item-Attribute-Affinity, los atributos de los datos de entrada no pueden incluir metadatos de elemento textuales no estructurados, como la descripción de un producto.
- Si utiliza un filtro con parámetros de marcadores de posición, debe incluir los valores de los parámetros en los datos de entrada de un objeto `filterValues`. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.
- Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente) que para sus datos de entrada.

Preparación de los datos de entrada para los segmentos de usuarios

Los trabajos de segmentos por lotes utilizan una versión de la solución para crear segmentos de usuarios en función de los datos que proporciona en un archivo JSON de entrada. Para poder obtener segmentos de usuarios, debe preparar y subir el archivo JSON a un bucket de Amazon S3. Le recomendamos que cree una carpeta de salida en su bucket de Amazon S3 o utilice un bucket de Amazon S3 de salida independiente. A continuación, puede ejecutar varios trabajos de inferencia por lotes utilizando la misma ubicación de datos de entrada.

Si utiliza un filtro con parámetros de marcadores de posición, por ejemplo, `$GENRE`, debe proporcionar los valores de los parámetros de un objeto `filterValues` en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.

Para preparar e importar datos

1. Aplique formato a sus datos de entrada del lote en función de la receta que utilice su solución. Separe el elemento de datos de entrada con una línea nueva. Los datos de entrada son una lista de itemIds (Item-Affinity) o atributos de elemento (Item-Attribute-Affinity).

   - Para atributos de elementos, los datos de entrada pueden incluir expresiones lógicas con el operador `AND` para obtener usuarios de varios elementos o atributos por consulta. Para obtener más información, consulte Especificación de atributos de elemento para la receta Item-Attribute-Affinity.
   - En el caso de los atributos de elemento, use el carácter \ para incluir caracteres especiales, apóstrofos o comillas en los datos de entrada.
   - Para ver ejemplos de datos de entrada para ambas recetas, consulte Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes.

2. Suba el JSON de entrada a una carpeta de entrada de su bucket de Amazon S3. Para obtener más información, consulte Subida de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.
3. Cree una ubicación independiente para sus datos de salida, ya sea una carpeta o un bucket de Amazon S3 diferente. Al crear una ubicación independiente para el JSON de salida, puede ejecutar varios trabajos de segmentos por lotes con la misma ubicación de datos de entrada.

Una vez que haya preparado los datos de entrada y los haya cargado en un bucket de Amazon S3, podrá generar segmentos de usuarios con un trabajo de segmentos por lotes. Para obtener más información, consulte Obtención de segmentos de usuario con un trabajo de segmentos por lotes.

Temas

- Especificación de atributos de elemento para la receta Item-Attribute-Affinity
- Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes

Especificación de atributos de elemento para la receta Item-Attribute-Affinity

Si utiliza la receta Item-Attribute-Affinity, los datos de entrada son una lista de atributos de elemento. Puede mezclar diferentes columnas de metadatos. Por ejemplo, una fila puede ser una columna numérica y la siguiente puede ser una columna categórica. No puede usar metadatos de elemento textuales no estructurados como un atributo de elemento.

Los metadatos de elemento de entrada pueden incluir expresiones lógicas con el operador `AND` para obtener un segmento de usuarios para varios atributos. Por ejemplo, una línea de sus datos de entrada puede ser `{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}` o `{"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.audience = "\teen\""}`.

Al combinar dos atributos con el operador `AND`, crea un segmento de usuario con los usuarios que tienen más probabilidades de interactuar con los elementos que tienen ambos atributos en función del historial de interacciones de usuarios. A diferencia de las expresiones de filtro (que utilizan el operador `IN` para igualdad de cadena), las expresiones de entrada de segmentos por lotes solo admiten el símbolo = para igualdad para la coincidencia de cadenas.

Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes

Para un trabajo de segmentos por lotes, los datos de entrada deben ser una lista de itemIds (receta Item-Affinity) o atributos de elemento (Item-Attribute-Affinity). Cada línea de datos de entrada es una consulta de inferencia independiente. Cada segmento de usuario se ordena en orden descendente en función de la probabilidad de que cada usuario interactúe con los elementos de su inventario.

Si utiliza un filtro con parámetros de marcadores de posición, por ejemplo, `$GENRE`, debe proporcionar los valores de los parámetros de un objeto `filterValues` en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.

A continuación se muestran ejemplos de entrada y salida JSON con el formato correcto para trabajos de segmentos por lotes organizados por receta.

Item-Affinity

Input

Los datos de entrada pueden tener un máximo de 500 elementos. Separe cada `itemId` con una nueva línea de la siguiente manera.

```
{"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"]}}
...
```

Item-Attribute-Affinity

Input

Los datos de entrada pueden tener un máximo de 10 consultas, cada una de las cuales consta de uno o más atributos de elemento no textuales. Separe cada atributo o expresión de atributo con una línea nueva de la siguiente manera.

```
{"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"]}}
...
```

Obtención de segmentos de usuario con un trabajo de segmentos por lotes

Si utilizó una receta USER_SEGMENTATION, puede crear trabajos de segmentos por lotes para obtener segmentos de usuarios con la versión de su solución. Cada segmento de usuario se ordena en orden descendente en función de la probabilidad de que cada usuario interactúe con los elementos de su inventario. Según la receta, los datos de entrada deben ser una lista de elementos (Receta Item-Affinity) o atributos de elementos (Receta Item-Attribute-Affinity) en formato JSON. Puede crear un trabajo de segmento por lotes con la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) o AWS SDKs.

Cuando crea un trabajo de segmento por lotes, especifica las rutas de Amazon S3 a sus ubicaciones de entrada y salida. Amazon S3 se basa en prefijos. Si proporciona un prefijo para la ubicación de los datos de entrada, Amazon Personalize utilizará todos los archivos que coincidan con ese prefijo como datos de entrada. Por ejemplo, si proporciona `s3://amzn-s3-demo-bucket/folderName` y su bucket también tiene una carpeta con una ruta de `s3://amzn-s3-demo-bucket/folderName_test`, Amazon Personalize utiliza todos los archivos de ambas carpetas como datos de entrada. Para usar solo los archivos de una carpeta específica como datos de entrada, finalice la ruta de Amazon S3 con un delimitador de prefijo, como `/`: `s3://amzn-s3-demo-bucket/folderName/` Para obtener más información sobre cómo Amazon S3 organiza los objetos, consulte Organizar, enumerar y trabajar con sus objetos.

Temas

- Creación de un trabajo de segmento por lotes (consola)

- Creación de un trabajo de segmento por lotes (AWS CLI)
- Creación de un trabajo de segmento por lotes (AWS SDKs)

Creación de un trabajo de segmento por lotes (consola)

Cuando haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de segmento por lotes. Este procedimiento supone que ya ha creado una solución y una versión de la solución (modelo entrenado) con una receta USER_SEGEMENTATION.

Para obtener un trabajo de segmento por lotes (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. En la página Grupo de conjuntos de datos, elija su grupo de conjuntos de datos.
3. Elija trabajos de inferencia por lotes en el panel de navegación y, a continuación, Crear trabajo de inferencia por lotes.
4. En Detalles del trabajo de segmento por lotes, para Nombre del trabajo de segmento por lotes, especifique un nombre para su trabajo de segmento por lotes.
5. En Solución, elija la solución y, a continuación, elija el ID de versión de la solución que desee utilizar para generar las recomendaciones. Puede crear trabajos de segmentos por lotes solo si utilizó una receta USER_SEGEMENTATION.
6. En Número de usuarios, si lo desea, especifique el número de usuarios que Amazon Personalize genera para cada segmento de usuarios. El valor predeterminado es 25. El máximo es de 5 millones.
7. En Origen de entrada, especifique la ruta de Amazon S3 al archivo de entrada o utilice Examinar S3 para elegir su bucket de Amazon S3.

   Utilice la siguiente sintaxis: **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`**

   Los datos de entrada deben tener el formato correcto para la receta que utiliza su solución. Para ver ejemplos de datos de entrada, consulte Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes.

8. En Destino de salida, especifique la ruta a su ubicación de salida o utilice Examinar S3 para elegir su bucket de Amazon S3. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente).

   Utilice la siguiente sintaxis: **`s3://amzn-s3-demo-bucket/<output folder name>/`**
9. En Rol de IAM, elija una de las opciones siguientes:

   - Elija Crear y usar un nuevo rol de servicio e introduzca el Nombre del rol de servicio para crear un nuevo rol, o
   - Si ya ha creado un rol con los permisos correctos, elija Usar un rol de servicio existente y elija el rol de IAM.

   El rol que utilice debe tener acceso de lectura y escritura a sus buckets de entrada y salida de Amazon S3, respectivamente.
10. Para Configuración de filtro, elija opcionalmente un filtro para aplicar un filtro a los segmentos de usuarios. Si el filtro utiliza parámetros de marcador de posición, asegúrese de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada.
11. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
12. Seleccione Crear trabajo de segmento por lotes. Se inicia la creación del trabajo de segmentos por lotes y aparece la página Trabajos de inferencia por lotes con la sección Detalles del trabajo de inferencia por lotes visible.
13. Cuando el estado del trabajo de segmentos por lotes cambia a Activo, puede recuperar la salida del trabajo en el bucket de Amazon S3 de salida designado. El nombre del archivo de salida tendrá el formato *`input-name`*`.out`.

Creación de un trabajo de segmento por lotes (AWS CLI)

Cuando haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de segmento por lotes con el siguiente código `create-batch-segment-job`. Especifique un nombre de trabajo, sustituya `Solution version ARN` por el nombre de recurso de Amazon (ARN) de la versión de su solución, y sustituya el `IAM service role ARN` por el ARN del rol de servicio de IAM que creó para Amazon Personalize durante la configuración. Este rol debe tener acceso de lectura y escritura a sus buckets de entrada y salida de

Amazon S3, respectivamente. Para `num-results`, especifique el número de usuarios que desea que Amazon Personalize prediga para cada línea de datos de entrada. El valor predeterminado es 25. El máximo es de 5 millones. Si lo desea, proporcione un `filter-arn` para filtrar segmentos de usuarios. Si el filtro utiliza parámetros de marcador de posición, asegúrese de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados).

Sustituya `S3 input path` y `S3 output path` por la ruta de Amazon S3 al archivo de entrada y a las ubicaciones de salida. Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice la siguiente sintaxis para las ubicaciones de entrada y salida: **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`** y **`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} \
                --role-arn IAM service role ARN
{
   "batchSegmentJobArn": "arn:aws:personalize:us-west-2:acct-id:batch-segment-job/
batchSegmentJobName"
}
```

Creación de un trabajo de segmento por lotes (AWS SDKs)

Cuando haya completado Preparación de los datos de entrada para las recomendaciones por lotes, estará listo para crear un trabajo de segmento por lotes con la operación `CreateBatchSegmentJob`. En el siguiente código se muestra cómo crear un trabajo de segmento por lotes. Asigne un nombre al trabajo, especifique el nombre de recurso de Amazon (ARN) de la versión de la solución que se utilizará, especifique el ARN para su rol de IAM de Amazon Personalize y especifique la ruta de Amazon S3 a sus ubicaciones de salida y archivo de entrada. Su rol de servicio de IAM debe tener acceso de lectura y escritura a los buckets de Amazon S3 de entrada y salida, respectivamente.

Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Utilice la siguiente sintaxis para las ubicaciones de entrada y

salida: **s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json** y **s3://amzn-s3-demo-bucket/<output folder name>/**.

Para numResults, especifique el número de usuarios que desea que Amazon Personalize prediga para cada línea de datos de entrada. El valor predeterminado es 25. El máximo es de 5 millones. Si lo desea, proporcione un filterArn para filtrar segmentos de usuarios. Si el filtro utiliza parámetros de marcador de posición, asegúrese de que los valores de los parámetros estén incluidos en el JSON de entrada. Para obtener más información, consulte Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados).

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();
```

El procesamiento del trabajo por lotes puede tardar un tiempo en completarse. Puede comprobar el estado de un trabajo llamando a DescribeBatchSegmentJob y pasando un `batchSegmentJobArn` como el parámetro de entrada. También puede publicar todos los trabajos del segmento por lotes de Amazon Personalize de su AWS entorno llamando al teléfonoListBatchSegmentJobs.

Ejemplos de formato de resultados de trabajos de segmento por lotes

Un trabajo de segmento por lotes importa los datos de entrada del lote desde un bucket de Amazon S3, utiliza la versión de la solución entrenada con una receta USER_SEGMENTATION para generar segmentos de usuarios y exporta los segmentos a un bucket de Amazon S3.

En las siguientes secciones se enumeran ejemplos de salidas JSON para tareas de segmentos por lotes por receta.

Temas

- Item-Affinity
- Item-Attribute-Affinity

Item-Affinity

En el ejemplo siguiente, se muestra el formato del archivo JSON de salida para una receta 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"]}}
...
```

Item-Attribute-Affinity

En el ejemplo siguiente, se muestra el formato del archivo JSON de salida para una receta Item-Attribute-Affinity.

```
{"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"]}}
...
```

Recomendaciones de filtrado y segmentos de usuarios

Al obtener recomendaciones con un recomendador de dominio o una campaña personalizada, puede filtrar los resultados según distintos criterios personalizados. Por ejemplo, es posible que no desee recomendar productos que un usuario ya haya comprado o recomendar solo elementos para un grupo de edad concreto.

Del mismo modo, con las recetas USER_SEGMENTATION, es posible que no desee incluir determinados tipos de usuarios en los segmentos de usuarios. Al filtrar los resultados, puede controlar los elementos que se recomendarán a los usuarios o los usuarios que se incluirán en los segmentos de usuarios.

Puede crear, editar, eliminar y aplicar filtros mediante la consola Amazon Personalize, el AWS Command Line Interface (AWS CLI) y los AWS SDK.

- Para obtener recomendaciones en tiempo real, debe aplicar un filtro y especificar los valores de los parámetros del filtro al llamar a las GetPersonalizedRanking operaciones GetRecommendations GetActionRecommendations, o. También puede aplicar un filtro al obtener recomendaciones de una campaña o un recomendador en la consola.

  Cuando reciba recomendaciones de elementos en tiempo real con recetas o casos de uso de elementos personalizados o relacionados, puede especificar una promoción en la solicitud. Una promoción usa un filtro para definir reglas empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados. Para obtener más información, consulte Promoción de elementos en las recomendaciones en tiempo real.
- En el caso de flujos de trabajo por lotes, incluya cualquier valor de parámetro de filtro en el JSON de entrada. A continuación, especifique el nombre de recurso de Amazon (ARN) del filtro al crear un trabajo de inferencia por lotes o un trabajo de segmento de lote. Para obtener más información, consulte Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados).

Filtrar actualizaciones para nuevos registros

Para los datos que importe con las PutActionInteractions operaciones PutEvents o, Amazon Personalize actualiza todos los filtros del grupo de conjuntos de datos con los nuevos datos en cuestión de segundos después de la importación. Por ejemplo, si tu filtro elimina los artículos comprados de las recomendaciones y registras un evento de compra para un usuario con la

PutEvents operación, este artículo se eliminará de las recomendaciones futuras para este usuario segundos después de registrar el evento.

Para el resto de los datos importados de forma masiva o individual, Amazon Personalize actualiza todos los filtros del grupo de conjuntos de datos con los nuevos datos en un plazo de 20 minutos a partir de la última importación.

Temas

- Expresiones de filtro
- Filtrado de recomendaciones en tiempo real
- Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados)

Expresiones de filtro

Para configurar filtros, debe usar una expresión de filtro con el formato correcto. Las expresiones de filtro se componen de identificadores de campo y conjuntos de datos en el formato `dataset.field`, junto con operadores lógicos, palabras clave y valores. En el caso de los valores, puede especificar valores fijos o añadir parámetros marcadores de posición para establecer los criterios de filtro cuando reciba recomendaciones.

Puede usar expresiones de filtro para filtrar elementos, usuarios o acciones de las recomendaciones en función de los datos de los siguientes conjuntos de datos:

- Interacciones de elemento: puede usar expresiones de filtro para incluir o excluir elementos o usuarios en función de los datos de interacciones. Por ejemplo, puede excluir los artículos en los que un usuario ya haya hecho clic (para obtener recomendaciones de artículos) o incluir solo a los usuarios que hayan valorado los artículos (para la Item-Affinity receta). Para todos los tipos de recetas, solo puede filtrar por el tipo de evento. No puede filtrar por otros metadatos de interacción, como los metadatos contextuales. No puede utilizar interacciones de elemento con la Receta Item-Attribute-Affinity.

  Amazon Personalize considera hasta 100 de las interacciones más recientes por usuario y por tipo de evento. Esta es una cuota ajustable. Puede solicitar un aumento de cuota mediante la consola de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

- Interacciones de acción: utilice expresiones de filtro para incluir o excluir acciones con las que un usuario ha interactuado en función del tipo de evento. Por ejemplo, puede excluir acciones que un usuario ya haya realizado. No puede filtrar por otros metadatos de interacción de acción.

  Amazon Personalize tiene en cuenta hasta 300 de las interacciones de acción más recientes por usuario y por tipo de evento. Esta es una cuota ajustable. Puede solicitar un aumento de cuota mediante la consola de Service Quotas.
- Elementos: utilice expresiones de filtro para incluir o excluir elementos en función de condiciones de elementos específicas. No puede usar filtros para incluir o excluir elementos en función de metadatos de elementos textuales no estructurados, como descripciones de productos. Si el caso de uso de tu dominio o tu receta personalizada generan recomendaciones de artículos relacionados, como la Similar-Items receta o el caso de uso del dominio More Like X, puedes usar expresiones de filtro para incluir o excluir artículos en función de las propiedades del artículo que especifiques en tu solicitud de recomendación.
- Usuarios: en el caso de las recomendaciones de elementos y acciones, si tiene un conjunto de datos de usuarios, puede excluir o incluir elementos o acciones en función de un valor `CurrentUser`. En el caso de las recomendaciones personalizadas, elementos populares y recomendaciones de acción, este es el usuario para el que recibe las recomendaciones. Para los elementos relacionados, este es un usuario opcional que puede especificar en la solicitud de recomendación.

  En el caso de los segmentos de usuarios, puede usar expresiones de filtro para incluir o excluir usuarios de los segmentos de usuarios en función de atributos, como `Users.MEMBERSHIP_STATUS`.
- Acciones: utilice expresiones de filtro para incluir o excluir acciones en función de condiciones de acción específicas. Amazon Personalize excluye acciones automáticamente en función de sus datos de `Action expiration timestamp` y `Repeat frequency`. No puede crear filtros personalizados adicionales que filtren en función de estos datos.

Para obtener una lista completa de elementos de expresiones de filtro, consulte Filtrar elementos de expresión. Para ver ejemplos de expresiones de filtro, consulte Ejemplos de expresiones de filtro.

Temas

- Directrices y requisitos
- Estructura y elementos de las expresiones de filtro
- Ejemplos de expresiones de filtro

Directrices y requisitos

Al crear una expresión de filtro, tenga en cuenta las directrices y requisitos siguientes:

- No puede usar filtros para incluir o excluir elementos en función de metadatos de elementos textuales no estructurados, como descripciones de productos.
- Si filtra en función de los datos de interacciones de elemento o acción, solo puede filtrar por el tipo de evento. No puede filtrar por otros metadatos de interacción, como los metadatos contextuales.
- Amazon Personalize solo ignora mayúsculas y minúsculas cuando coincide con tipos de eventos.
- No puede usar conjuntos de datos de interacciones de elementos y de elementos en una sola expresión. Para crear un filtro que filtre por conjuntos de datos de interacciones y luego por conjuntos de datos de elementos (o lo contrario), debe encadenar dos o más expresiones. Para obtener más información, consulte Combinación de varias expresiones.
- No puede usar conjuntos de datos de interacciones de elementos y de acciones en una sola expresión. Para crear un filtro que filtre por interacciones de elementos y luego por conjuntos de datos de acciones (o lo contrario), debe encadenar dos o más expresiones. Para obtener más información, consulte Combinación de varias expresiones.
- No puede utilizar interacciones de elemento con la Receta Item-Attribute-Affinity.
- No puede crear expresiones de filtro que filtren utilizando valores de tipo booleano en su esquema. Para filtrar en función de valores booleanos, utilice un esquema con un campo de tipo String y utilice los valores `"True"` y `"False"` en sus datos. También puede usar el tipo int o long, y los valores `0` y `1`.
- El número máximo de campos de conjunto de datos distintos para un filtro, ya sea en una expresión o en varias expresiones encadenadas, es 10. El número máximo de campos de conjunto de datos distintos en todos los filtros de un grupo de conjuntos de datos es 20.
- Puedes aplicar un filtro con el CurrentItem elemento solo si el caso de uso del dominio o la receta personalizada generan recomendaciones de artículos relacionados, como la Similar-Items receta o el caso de uso del dominio More Like X.
- No puede usar parámetros de marcadores de posición en una expresión de filtro que utilice el operador NOT_IN. En su lugar, utilice el operador IN y la acción opuesta. Por ejemplo, utilice Incluir en lugar de Excluir (o al revés).
- No puede crear filtros que filtren en función de los datos de `Action expiration timestamp` y `Repeat frequency`. Amazon Personalize filtra las recomendaciones de acción de forma automática en función de estos datos.

Estructura y elementos de las expresiones de filtro

Esta sección incluye información sobre de la estructura de las expresiones de filtro y sus elementos.

Temas

- Estructura de las expresiones de filtro
- Filtrar elementos de expresión

Estructura de las expresiones de filtro

La estructura general de una expresión de filtro es la siguiente:

```
EXCLUDE/INCLUDE ItemID/ActionID/UserID WHERE dataset type.field IN/NOT IN (value/
parameter)
```

Puede crear expresiones de filtro manualmente u obtener ayuda con la estructura y la sintaxis de expresiones mediante el Generador de expresiones de la consola.

Filtrar elementos de expresión

Utilice los siguientes elementos para crear expresiones de filtro:

INCLUDE o EXCLUDE

Utilice `INCLUDE` para limitar las recomendaciones a solo los elementos que cumplen los criterios de filtro O use `EXCLUDE` para eliminar todos los elementos que cumplen los criterios de filtro.

ItemID/ActionID/UserID

Use uno de estos elementos después del elemento `INCLUDE` o `EXCLUDE`. El elemento que utilice depende de si está filtrando elementos (para recomendaciones de elementos), acciones (para recomendaciones de acciones) o usuarios (para segmentos de usuarios).

WHERE

Utilice `WHERE` para comprobar las condiciones de los elementos, acciones o usuarios. Debe usar el elemento `WHERE` después de `ItemID`, `ActionID` o `UserID`.

AND/OR

Para encadenar varias condiciones dentro de la misma expresión de filtro, utilice `AND` o `OR`. Las condiciones encadenadas mediante `AND` o `OR` solo pueden afectar a los campos del conjunto de datos utilizados en la primera condición.

Dataset.field

Proporcione el conjunto de datos y el campo de metadatos por los que desea filtrar las recomendaciones con el formato `dataset.field`. Por ejemplo, para filtrar las recomendaciones de artículos en función del campo de géneros de tu conjunto de datos de artículos, Items.genres utilizarías una expresión de filtro.

Condición IF

Utilice una condición `IF` solo para comprobar las condiciones de una expresión `CurrentUser` y solo una vez al final de la misma. Sin embargo, puede extender una condición `IF` mediante `AND`.

CurrentUser.attribute

Para filtrar las recomendaciones de elementos en función del usuario para el que está recibiendo recomendaciones, en solo una condición IF, utilice `CurrentUser` y proporcione el campo de usuario. Por ejemplo, `CurrentUser.AGE`.

CurrentItem.attribute

Para casos de uso y recetas de elementos relacionados, utilice `CurrentItem.attribute` para filtrar elementos en función de un atributo del elemento que especifique en su solicitud de recomendaciones de elementos relacionados. Por ejemplo, `CurrentItem.GENRE` o `CurrentItem.PRICE`.

Puedes aplicar un filtro con el CurrentItem elemento solo si el caso de uso de tu dominio o receta personalizada genera recomendaciones de artículos relacionados, como la Similar-Items receta o el caso de uso del dominio More Like X. La primera vez que cree un filtro con un elemento `CurrentItem`, la creación del filtro tardará unos minutos. Si lo utilizas AWS KMS para cifrar, la creación del filtro puede tardar hasta 15 minutos.

IN/NOT EN

Utilice `IN` o `NOT IN` como operadores de comparación para filtrar en función de la coincidencia (o no coincidencia) de uno o más valores de cadena. Amazon Personalize filtra solo las cadenas exactas.

Operadores de comparación

Utilice los operadores =, <, <=, >, >= y != para probar la igualdad de los datos numéricos, incluidos los datos pasados en un parámetro de marcador de posición.

Carácter de asterisco (*)

Use * para incluir o excluir interacciones de todo tipo. Use * solo para filtrar expresiones que empleen el campo `EVENT_TYPE` de un conjunto de datos de `Interactions`.

Separador vertical

Utilice el separador vertical (|) para encadenar varias expresiones. Para obtener más información, consulte Combinación de varias expresiones.

Parameters

En el caso de las expresiones que utilizan operadores de comparación o el operador `IN`, utilice el signo de dólar ($) y un nombre de parámetro para añadir un parámetro de marcador de posición como valor. Por ejemplo, `$GENRES`. En este ejemplo, cuando reciba recomendaciones, especifica los géneros por los que desea filtrar.

> **Note**
>
> El nombre de un parámetro se define al añadirlo a una expresión. El nombre del parámetro no tiene que coincidir con el nombre del campo. Recomendamos que utilice un nombre de parámetro que sea similar al nombre del campo y sencillo de recordar. Utilice el nombre del parámetro (distingue entre mayúsculas y minúsculas) al aplicar el filtro a las solicitudes de recomendaciones. Para ver un ejemplo que muestra cómo aplicar un filtro con parámetros de marcador de posición cuando se utilizan los AWS SDK, consulte. Aplicar un filtro (AWS SDK)

Ejemplos de expresiones de filtro

Use las expresiones de filtro de las siguientes secciones para aprender a crear sus propias expresiones de filtro.

Temas

- Ejemplos de expresiones de filtro de recomendaciones de elementos

- Expresiones de filtro de segmentos de usuarios
- Ejemplos de expresiones de filtro de recomendaciones de acciones
- Combinación de varias expresiones

Ejemplos de expresiones de filtro de recomendaciones de elementos

Las siguientes expresiones de filtro muestran cómo filtrar las recomendaciones de elementos en función de las interacciones y los metadatos de los elementos, así como los metadatos de usuarios. Están organizados por tipo de datos.

Temas

- Datos de interacciones de elementos
- Datos de elementos
- Datos de usuario

Datos de interacciones de elementos

La siguiente expresión excluye elementos en función de un tipo de evento (como hacer clic) o de los tipos de eventos que se especifican al recibir recomendaciones mediante el parámetro `$EVENT_TYPE`.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

La siguiente expresión excluye los elementos en los que un usuario ha hecho clic o ha transmitido en streaming.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click", "stream")
```

La siguiente expresión solo incluye los elementos en los que el usuario ha hecho clic.

```
INCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click")
```

Datos de elementos

La siguiente expresión excluye los elementos en función de una o varias categorías que especifique al obtener recomendaciones mediante el parámetro `$CATEGORY`.

```
EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)
```

La siguiente expresión incluye solo los elementos que son más baratos que el elemento actual (el elemento que especifica en la solicitud para recomendaciones de elementos relacionados) y que se han creado por el mismo estudio que el elemento actual. Puedes aplicar un filtro con el CurrentItem elemento solo si el caso de uso de tu dominio o tu receta personalizada generan recomendaciones de artículos relacionados.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.GENRE IN
 CurrentItem.GENRE
```

La siguiente expresión excluye elementos basados en varios niveles de campos categóricos. Excluye los elementos con un valor de CATEGORY_L1 `shoe` que no tienen un valor de CATEGORY_L2 de `boot`.

```
EXCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("shoe") AND Items.CATEGORY_L2 NOT IN
 ("boot")
```

La siguiente expresión incluye solo los elementos con un precio inferior o igual que el precio que especifica al recibir recomendaciones mediante el parámetro `$PRICE`.

```
INCLUDE ItemID WHERE Items.PRICE <= $PRICE
```

La siguiente expresión solo incluye elementos que se han creado antes de una marca de tiempo (en tiempo Unix) que especifica al recibir recomendaciones.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

La siguiente expresión solo incluye elementos con un género o géneros que especifica al recibir recomendaciones con el parámetro `$GENRE`.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE)
```

La siguiente expresión solo incluye elementos que son más caros que el elemento actual y que se han creado más recientemente que la marca temporal (en tiempo Unix) que especifique. Puede utilizar este filtro si recibe recomendaciones de elementos relacionados y desea aplicar algunas reglas empresariales específicas basadas en el precio y en una fecha de creación variable.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.CREATION_TIMESTAMP >
 $DATE
```

Datos de usuario

La siguiente expresión excluye los elementos de un género o géneros que especifique al recibir recomendaciones mediante el parámetro $GENRE, pero solo si la edad del usuario actual es igual al valor que especifique al obtener recomendaciones mediante el parámetro $AGE.

```
EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE) IF CurrentUser.AGE = $AGE
```

La siguiente expresión solo incluye elementos con `watch` para CATEGORY_L1 y `luxury` para CATEGORY_L2, si la edad del usuario actual es mayor que 18.

```
INCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("watch") AND Items.CATEGORY_L2 IN ("luxury")
 IF CurrentUser.AGE > 18
```

Expresiones de filtro de segmentos de usuarios

Las siguientes expresiones de filtro muestran cómo filtrar los segmentos de usuarios en función de los datos de interacciones de elementos y los metadatos de usuarios. Están organizados por tipo de datos.

Datos de usuario

La siguiente expresión de filtro incluye solo los usuarios con un estado de pertenencia igual al valor que especifique al obtener los segmentos de usuarios.

```
INCLUDE UserID WHERE Users.MEMBERSHIP_STATUS IN ($MEMBERSHIP)
```

La siguiente expresión de filtro excluye a los usuarios con un valor AGE inferior al que especifique al obtener segmentos de usuarios.

```
EXCLUDE UserID WHERE Users.AGE < $AGE
```

Datos de interacciones de elementos

La siguiente expresión de filtro solo incluye a los usuarios que han hecho clic en los elementos o los han valorado.

```
INCLUDE UserID WHERE Interactions.EVENT_TYPE IN ("click", "rating")
```

La siguiente expresión de filtro excluye usuarios de los segmentos de usuarios que tienen interacciones de elementos con un tipo de evento que especifique al obtener segmentos de usuarios.

```
EXCLUDE UserID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

Ejemplos de expresiones de filtro de recomendaciones de acciones

Los siguientes ejemplos de expresiones de filtro muestran cómo filtrar las acciones en función de los datos de interacciones de las acciones, los datos de las acciones y los datos de los usuarios. Están organizados por tipo de datos.

Temas

- Datos de interacciones de acción
- Datos de acciones
- Datos de usuario

Datos de interacciones de acción

La siguiente expresión de filtro incluye solo las acciones de las recomendaciones con las que el usuario ha interactuado, cuando estas interacciones tengan un tipo de evento que especifique al recibir recomendaciones.

```
INCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

La siguiente expresión de filtro excluye las acciones que el usuario no haya realizado en función del tipo de evento.

```
EXCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ("NOT_TAKEN")
```

Datos de acciones

La siguiente expresión excluye acciones en función de una o varias categorías que especifique al obtener recomendaciones mediante el parámetro `$CATEGORY`.

```
EXCLUDE ActionID WHERE Actions.CATEGORY IN ($CATEGORY)
```

La siguiente expresión solo incluye las acciones con un valor superior al que se especifica al recibir recomendaciones.

```
INCLUDE ActionID WHERE Actions.VALUE > ($VALUE)
```

Datos de usuario

La siguiente expresión solo incluye acciones de los miembros premium si el usuario actual tiene una suscripción premium.

```
INCLUDE ActionID WHERE Action.MEMBERSHIP_LEVEL IN ("Premium") IF CurrentUser.MEMBERSHIP
 = $PREMIUM
```

La siguiente expresión excluye acciones con un valor VALUE inferior al que especifica al recibir recomendaciones si el usuario actual es un miembro premium.

```
EXCLUDE ActionID WHERE Actions.VALUE < ($VALUE) IF CurrentUser.MEMBERSHIP = $PREMIUM
```

Combinación de varias expresiones

Para combinar varias expresiones juntas, utilice un separador vertical (`|`). Use una combinación de expresiones cuando quiera usar un solo filtro y filtrar los conjuntos de datos de elementos y de interacciones de elementos, o bien los conjuntos de datos de acciones e interacciones de acción. Cada expresión se evalúa primero de manera independiente y el resultado es la unión o la intersección de los dos resultados. Los siguientes ejemplos muestran cómo crear expresiones para conjuntos de datos de elementos y de interacciones de elementos, si bien se aplican las mismas reglas al trabajar con acciones e interacciones de acción.

Ejemplo de expresiones coincidentes

Si ambas expresiones utilizan `EXCLUDE` o `INCLUDE`, el resultado es la unión de los dos resultados de la siguiente manera (A y B son expresiones diferentes):

- `Exclude A | Exclude B` es igual a `Exclude result from A or result from B`
- `Include A | Include B` es igual a `Include result from A or result from B`

En el ejemplo siguiente se muestra cómo utilizar dos expresiones que usan `INCLUDE`. La primera expresión solo incluye los elementos con una o varias categorías que especifica al obtener

recomendaciones mediante el parámetro `$CATEGORY`. La segunda expresión incluye los elementos que el usuario ha marcado como `favorite`. Las recomendaciones solo incluirán los elementos con la categoría que especifique, junto con los elementos que el usuario haya marcado como favoritos.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY) | INCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("favorite")
```

Ejemplo de INCLUDE y EXCLUDE

Si una o más expresiones utilizan `INCLUDE` y una expresión utiliza `EXCLUDE`, el resultado es la resta del resultado de la expresión `EXCLUDE` del resultado de la expresión `INCLUDE` de la siguiente manera (A, B, C y D son expresiones diferentes).

- `Include A | Exclude B` es igual a `Include result from A - result from B`
- `Include A | Include B | Exclude C | Exclude D` es igual a `Include (A or B) - (C or D)`

El orden de las expresiones no importa: si la expresión EXCLUDE se encuentra antes que la expresión INCLUDE, el resultado es el mismo.

El siguiente ejemplo muestra cómo combinar una expresión `INCLUDE` y una expresión `EXCLUDE`. La primera expresión incluye solo los elementos de un género o géneros que especifica al recibir recomendaciones mediante el parámetro `$GENRE`. La segunda expresión excluye los elementos en los que el usuario ha hecho clic o ha transmitido en streaming. Las recomendaciones incluirán solo los elementos con un género que especifique y que no hayan recibido clics ni se hayan reproducido en streaming.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE) | EXCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("click", "stream")
```

Filtrado de recomendaciones en tiempo real

Puede filtrar las recomendaciones en tiempo real con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o los AWS SDK.

Al obtener recomendaciones de elementos personalizadas o de elementos similares, puede especificar una promoción en la solicitud. Una promoción usa un filtro para definir reglas

empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados. Para obtener más información, consulte Promoción de elementos en las recomendaciones en tiempo real.

Temas

- Filtrado de recomendaciones en tiempo real (consola)
- Filtrar recomendaciones en tiempo real (AWS CLI)
- Filtrar recomendaciones en tiempo real (AWS SDK)

Filtrado de recomendaciones en tiempo real (consola)

Para filtrar recomendaciones en tiempo real, cree un filtro y, a continuación, aplíquelo a una solicitud de recomendación.

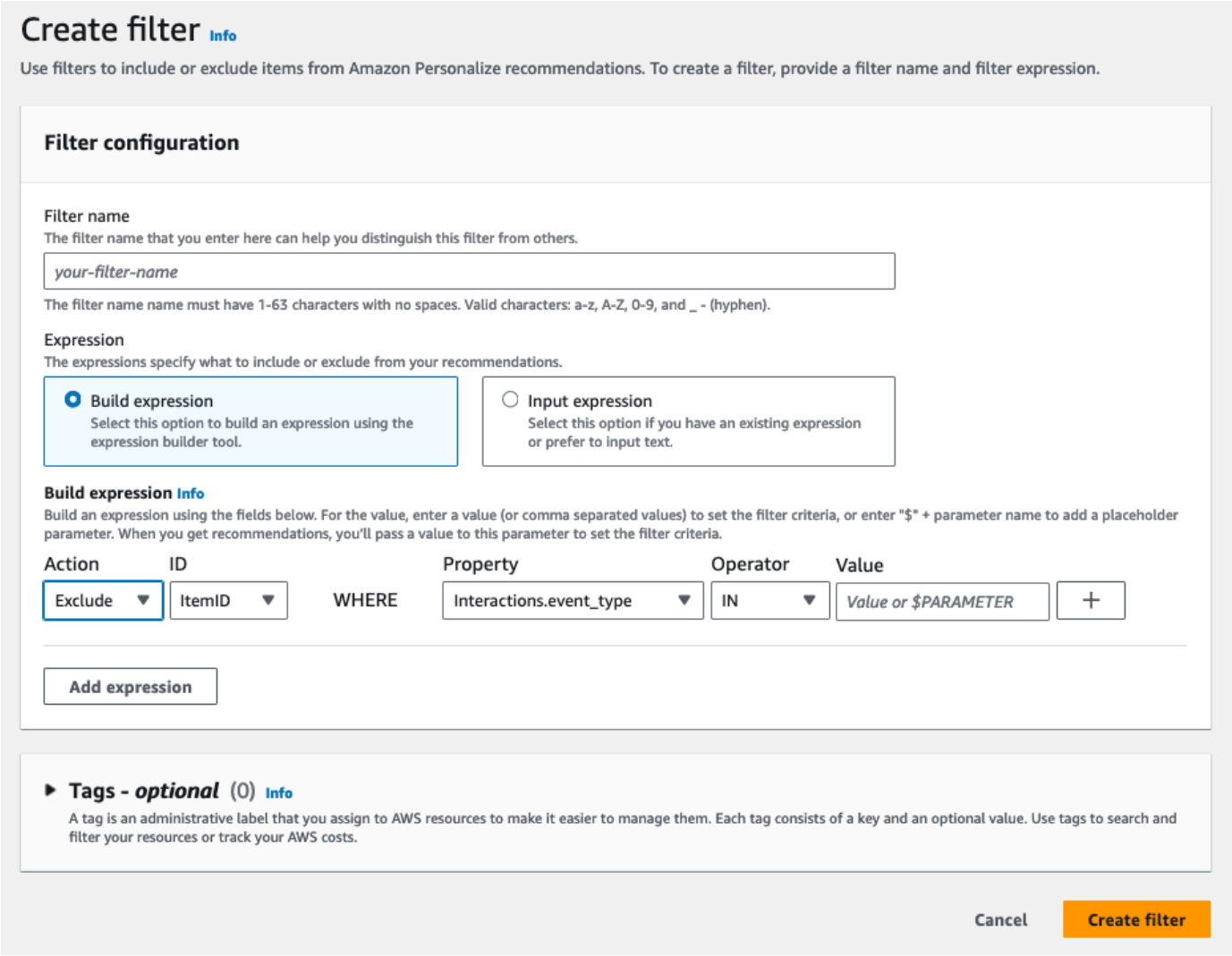
> **Note**
>
> Para filtrar recomendaciones mediante un filtro con parámetros y una campaña implementada antes del 10 de noviembre de 2020, debe volver a implementar la campaña mediante la operación UpdateCampaign o crear una nueva campaña.

Creación de un filtro (consola)

Para crear un filtro en la consola, elige el grupo de conjuntos de datos que contiene la campaña o el generador de recomendaciones que desea usar para obtener recomendaciones filtradas. A continuación, proporcione un nombre de filtro y una expresión de filtro.

Para crear un filtro (consola)

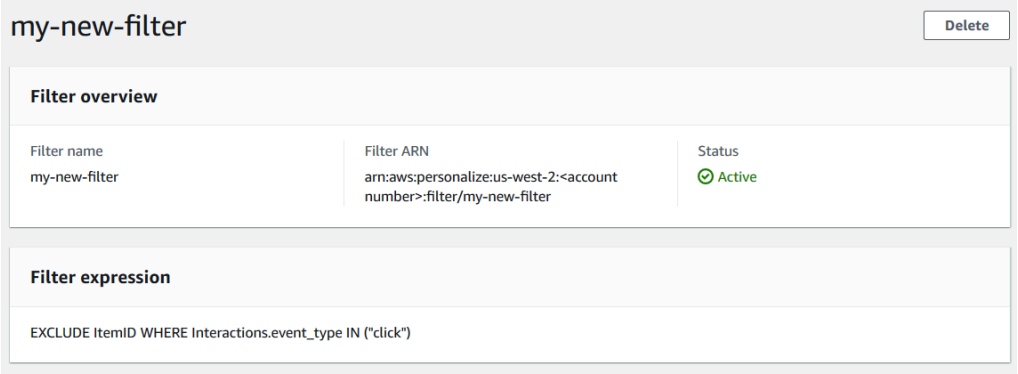
1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija el grupo de conjuntos de datos que contiene la campaña o el generador de recomendaciones que desee usar para obtener recomendaciones filtradas.
3. En la página de navegación, elija Filtros y, a continuación, Crear filtro. Aparece la página Crear filtro.



4. Para Nombre del filtro, introduzca un nombre para el filtro. Elegirá el filtro con este nombre cuando lo aplique a una solicitud de recomendación.
5. En Expresión, elija Crear expresión o Añadir expresión manualmente, y cree o inserte la expresión:

   - Para usar el generador de expresiones, elija Crear expresión. El generador de expresiones proporciona la estructura, los campos y las directrices para crear expresiones de filtro con el formato correcto. Para obtener más información, consulte Uso del generador de expresiones de filtro.
   - Para introducir su propia expresión, elija Añadir expresión manualmente. Para obtener más información, consulte Filtrar elementos de expresión.
6. Seleccione Finalizar. La página de información general del filtro muestra el nombre de recurso de Amazon (ARN), el estado y la expresión completa del filtro. Para eliminar el filtro, elija

Eliminar. Para obtener información acerca de la búsqueda y eliminación de filtros después de haber salido de la página de información general, consulte Eliminación de un filtro (consola).



Aplicación de un filtro (consola)

Para aplicar un filtro, en Probar recomendador (para recomendadores) o Probar resultados de la campaña (para campañas personalizadas), elija el filtro y escriba los valores de los parámetros de filtro. A continuación, obtenga recomendaciones para un usuario.

> **Important**
>
> En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE`, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un elemento `EXCLUDE`, puede omitir `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

Para aplicar un filtro (consola)

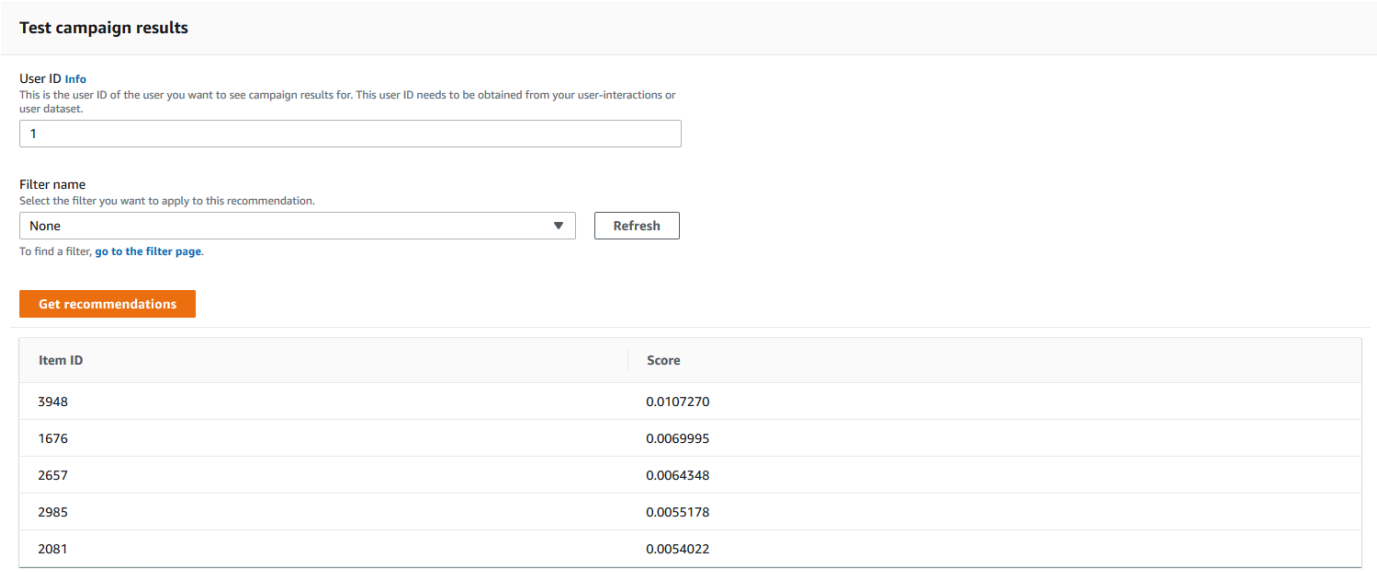
1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija el grupo de conjuntos de datos que contiene la campaña o el generador de recomendaciones que desee usar para obtener recomendaciones filtradas.
3. En función del tipo de grupo de conjunto de datos o el tipo de recurso, puede seguir uno de los procedimientos a continuación:

   a. Para un grupo de conjuntos de datos de dominio, en el panel de navegación, elija Generadores de recomendaciones.

b. Para un grupo de conjuntos de datos personalizados o recursos personalizados, en el panel de navegación, elija Recursos personalizados y, a continuación, Campañas.

4. En la página Generadores de recomendaciones o Campañas, elija el generador de recomendaciones o la campaña objetivo.

5. Para comparar, comience por obtener recomendaciones sin aplicar un filtro. En Probar recomendador/Probar resultados de la campaña, introduzca el ID de un usuario para el que quiera recibir recomendaciones, o el ID del elemento para obtener elementos relacionados, y elija Obtener recomendaciones. Aparece una tabla que contiene las recomendaciones principales.



6. En Nombre del filtro, elija el filtro que ha creado. Si el filtro tiene parámetros de marcador de posición, aparecen los campos asociados para cada parámetro.

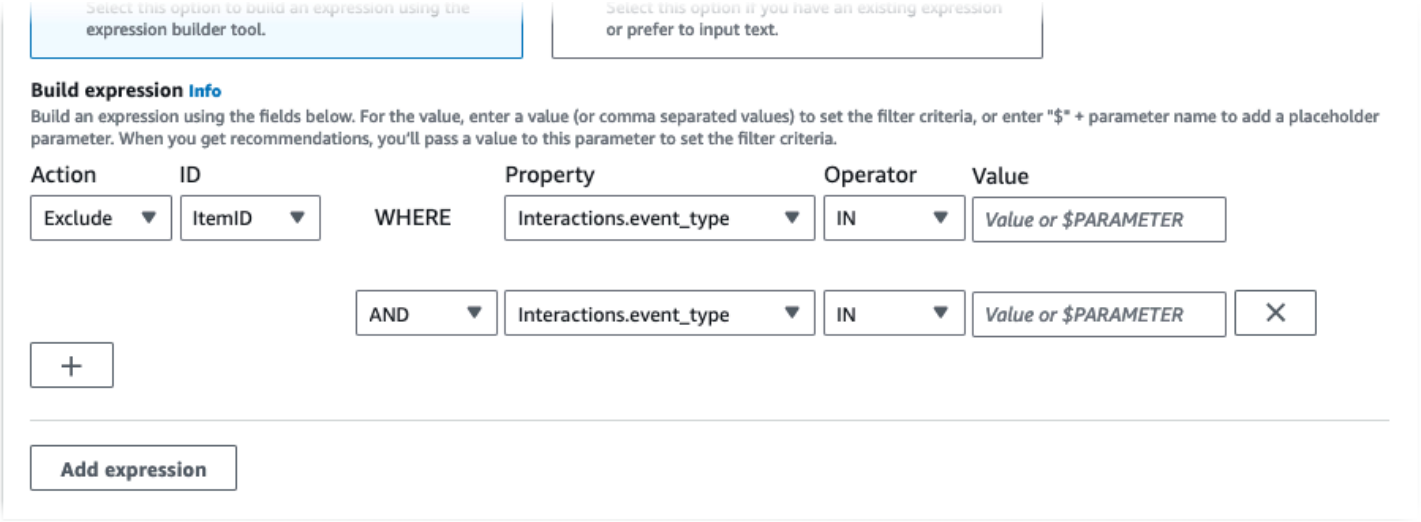
7. Si utiliza un filtro con parámetros de marcador de posición, introduzca el valor para establecer los criterios de filtro. Para usar varios valores para un parámetro, separe cada valor con una coma.

8. Con el mismo `User ID` o `Item ID` que en el paso anterior, elija Obtener recomendaciones. Aparece la tabla de recomendaciones.

   Por ejemplo, si el usuario ya ha comprado un producto recomendado, el filtro lo quita de la lista de recomendaciones. En este ejemplo, los elementos 2657 y 2985 se sustituyeron por los elementos más adecuados que el usuario no compró (los elementos 2641 y 1573).

Uso del generador de expresiones de filtro

El Generador de expresiones de la página Crear filtro proporciona la estructura, los campos y las directrices para crear un filtro con el formato correcto.



Para crear una expresión de filtro:

- Utilice los campos Tipo, Acción, Propiedad, Operador y Valor para crear una expresión.

  Para el Valor, introduzca un valor fijo o, para establecer los criterios de filtro cuando reciba recomendaciones, introduzca $ + un nombre de parámetro. Por ejemplo, `$GENRES`. Cuando reciba recomendaciones, proporcionará el valor o valores por los que filtrar. En este ejemplo, proporcionaría un género o una lista de géneros al recibir recomendaciones.

  Separe los valores que no sean parámetros con una coma. No puede añadir parámetros separados por comas a un filtro.

  > **Note**
  >
  > Tras elegir una Propiedad (en formato `dataset.field`), el valor de la Propiedad de las filas sucesivas encadenadas por las condiciones `AND` o `OR` debe ser el mismo `dataset`.

- Utilice los botones + y X para añadir o eliminar una fila de la expresión. No se puede eliminar la primera fila.
- Para las filas nuevas, utilice los operadores `AND`, `IF` o `OR` o del menú AND para crear una cadena de condiciones.

Para las condiciones `IF`:

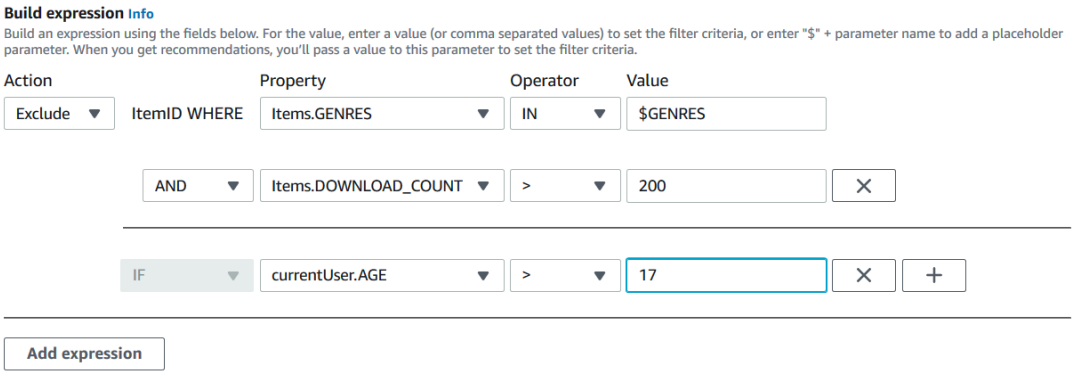
- Cada expresión solo puede contener un elemento `IF`. Si elimina una condición IF, el Generador de expresiones elimina cualquier condición `AND` posterior a la misma.
- Solo puede usar las condiciones `IF` para las expresiones que filtran por el `CurrentUser`.

- Elija el botón Agregar expresión para agregar una expresión de filtro adicional para un filtrado más preciso. Cada expresión se evalúa primero de forma independiente y el resultado es una unión de los dos resultados.

> **Note**
>
> Para crear un filtro que utilice conjuntos de datos tanto de elementos como de interacciones de elementos, o bien conjuntos de datos de acciones y de interacciones de acción, debe utilizar varias expresiones.

Ejemplo de generador de expresiones

En el siguiente ejemplo, se muestra cómo crear un filtro que excluya los elementos de un género que especifique al recibir recomendaciones (observe el parámetro de marcador de posición $GENRES). El filtro también excluye los elementos con un valor de `DOWNLOAD_COUNT` superior a `200`, pero solo si la edad del usuario actual es superior a 17.



Eliminación de un filtro (consola)

Al eliminar un filtro, se elimina el filtro de la lista de filtros de un grupo de conjuntos de datos.

> **Important**
>
> No puede eliminar un filtro mientras se está realizando un trabajo de inferencia por lotes.

Para eliminar un filtro (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En la lista de Grupos de conjuntos de datos, elija el grupo de conjuntos de datos que contiene el filtro que desea eliminar.
3. En el panel de navegación, elija Filtros.
4. En la lista de filtros, elige el filtro que desea eliminar y Ver detalles. Se abrirá la página de detalles del filtro.
5. Elija Eliminar y confirme la eliminación en el cuadro de diálogo de confirmación.

Filtrar recomendaciones en tiempo real (AWS CLI)

Para filtrar las recomendaciones mediante el AWS CLI, debe crear un filtro y, a continuación, aplicarlo especificando el ARN del filtro en una solicitud GetRecommendations oGetPersonalizedRanking.

> **Important**
>
> Para filtrar recomendaciones mediante un filtro con parámetros y una campaña que haya implementado antes del 10 de noviembre de 2020, debe volver a implementar la campaña mediante la llamada a UpdateCampaign o crear una nueva campaña.

Crear un filtro (AWS CLI)

Utilice la siguiente operación `create-filter` para crear un filtro y especificar la expresión de filtro.

Sustituya el `Filter name` por el nombre del filtro y el `Dataset group ARN` por el nombre de recurso de Amazon (ARN) del grupo de conjunto de datos. Sustituya el `filter-expression` de ejemplo por su propia expresión de filtro.

```
aws personalize create-filter \
  --name Filter name \
  --dataset-group-arn dataset group arn \
  --filter-expression "EXCLUDE ItemID WHERE Items.CATEGORY IN (\"$CATEGORY\")"
```

Si se realiza correctamente, aparecerá el ARN de filtro. Regístrelo para su uso posterior. Para comprobar que el filtro está activo, utilice la operación DescribeFilter antes de utilizar el filtro.

Para obtener más información sobre la API, consulte CreateFilter. Para obtener más información sobre las expresiones de filtro, incluidos ejemplos, consulte Estructura y elementos de las expresiones de filtro.

Aplicar un filtro (AWS CLI)

Cuando utilice las operaciones `get-recommendations`, `get-action-recommendations` o `get-personalized-ranking`, aplique un filtro al pasar el valor `filter-arn` y cualquier valor de filtro como parámetro.

A continuación se muestra un ejemplo de la operación `get-recommendations`. Sustituya `Campaign ARN` por el nombre de recurso de Amazon (ARN) de su campaña `User ID` por el ID del usuario para el que recibes recomendaciones, y `Filter ARN` por el ARN de su filtro. Si recibe recomendaciones de un generador de recomendaciones en lugar de una campaña, utilice `recommender-arn` en lugar de `--campaign-arn` y proporcione el ARN del generador de recomendaciones.

Si su expresión tiene algún parámetro, incluya el objeto `filter-values`. Para cada parámetro de su expresión de filtro, proporcione el nombre del parámetro (distingue entre mayúsculas y minúsculas) y los valores. Por ejemplo, si la expresión de filtro tiene un parámetro `$GENRE`, escriba “GENRE” como la clave, y uno o varios géneros, por ejemplo, `"Comedy"`, como el valor. Separe los valores con comas. Por ejemplo, `"\"comedy\",\"drama\",\"horror\"\"`.

> **Important**
>
> En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir elementos, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un elemento `EXCLUDE` para excluir elementos, puede omitir `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

```
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\""
    }'
```

Eliminar un filtro (AWS CLI)

Utilice la siguiente operación `delete-filter` para eliminar un filtro. Sustituya `filter ARN` por el ARN del filtro.

```
aws personalize delete-filter --filter-arn Filter ARN
```

Filtrar recomendaciones en tiempo real (AWS SDK)

Para filtrar las recomendaciones mediante los AWS SDK, debe crear un filtro y, a continuación, aplicarlo especificando el ARN del filtro en GetRecommendations una GetPersonalizedRanking solicitud o.

> **Important**
>
> Para filtrar recomendaciones mediante un filtro con parámetros y una campaña que haya implementado antes del 10 de noviembre de 2020, debe volver a implementar la campaña mediante la llamada a UpdateCampaign o crear una nueva campaña.

Crear un filtro (AWS SDK)

Cree un filtro nuevo con la operación CreateFilter. El código siguiente muestra cómo crear un filtro. Especifique el nombre de filtro, el nombre de recurso de Amazon (ARN) de su grupo de conjunto de datos, y proporcione la expresión de filtro.

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();
```

Registre el ARN del filtro para su uso posterior. Para comprobar que el filtro está activo, utilice la operación DescribeFilter antes de utilizar el filtro. Para obtener más información sobre la API, consulte CreateFilter. Para obtener más información sobre las expresiones de filtro, incluidos ejemplos, consulte Estructura y elementos de las expresiones de filtro.

Aplicar un filtro (AWS SDK)

Cuando utilice las GetPersonalizedRanking operaciones GetRecommendations GetActionRecommendations, o, aplique un filtro pasando a `filterArn` y cualquier valor del filtro como parámetros.

El siguiente código muestra cómo obtener recomendaciones de elementos de Amazon Personalize filtradas para un usuario. Especifique el ID del usuario para el que desea obtener recomendaciones, el nombre de recurso de Amazon (ARN) de la campaña y el ARN de su filtro. Si recibe recomendaciones de un generador de recomendaciones en lugar de una campaña, utilice `recommenderArn` en lugar de `campaignArn` y proporcione el ARN del generador de recomendaciones.

Para `filterValues`, para cada parámetro opcional de la expresión de filtro, proporcione el nombre del parámetro (distingue entre mayúsculas y minúsculas) y el valor o los valores. Por ejemplo, si la expresión de filtro tiene un parámetro `$GENRES`, indique "GENRES" como la clave y uno o varios géneros, como `"\"Comedy"\"`, como valor. Para varios valores, separe cada valor con una coma. Por ejemplo, `"\"comedy\",\"drama\",\"horror\""`.

> **Important**
>
> En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir elementos, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un elemento `EXCLUDE` para excluir elementos, puede omitir `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

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

En el siguiente ejemplo, se utilizan dos parámetros, uno con dos valores y otro con uno. Según la expresión de filtro, modifique el código para añadir o eliminar los campos parameterName y 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();
```

Eliminar un filtro (AWS Python (SDK)

Utilice el siguiente método `delete_filter` para eliminar un filtro. Sustituya `filter ARN` por el ARN del filtro.

```
import boto3
personalize = boto3.client("personalize")

response = personalize.delete_filter(
  filterArn = "filter ARN"
)
```

Filtrado de las recomendaciones por lotes y segmentos de usuarios (recursos personalizados)

El filtrado de recomendaciones por lotes y segmentos de usuarios funciona casi igual que el filtrado de recomendaciones en tiempo real. Sigue el mismo flujo de trabajo descrito en Obtención de recomendaciones de elemento por lotes o Obtención de segmentos de usuarios por lotes. Para filtrar recomendaciones por lotes o segmentos de usuarios, haga lo siguiente:

1. Cree un filtro como lo haría para las recomendaciones en tiempo real. Para obtener más información, consulte Filtrado de recomendaciones en tiempo real.
2. Prepare los datos de entrada y súbalos a Amazon S3 tal y como se describe en Preparación de los datos de entrada para las recomendaciones por lotes o Preparación de los datos de entrada para los segmentos de usuarios. Si el filtro utiliza parámetros de marcadores de posición, debe añadir un objeto `filterValues` adicional. Para obtener más información, consulte Proporcionar valores de filtro en su JSON de entrada. Si el filtro no usa parámetros de marcadores de posición, los datos de entrada pueden seguir los ejemplos de Ejemplos de JSON de entrada y salida de trabajos de inferencia por lotes Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes
3. Cree una ubicación independiente para sus datos de salida, ya sea una carpeta o un bucket de Amazon S3 diferente.
4. Cree un trabajo de inferencia por lotes o un trabajo de segmento por lotes. Al crear el volumen, especifique el nombre de recurso de Amazon (ARN) del filtro.
5. Cuando se complete el trabajo segmento de lotes o inferencia por lotes, recupere las recomendaciones o los segmentos de usuarios de su ubicación de salida en Amazon S3.

Temas

- Proporcionar valores de filtro en su JSON de entrada
- Filtrado de flujos de trabajo por lotes (consola)
- Filtrar flujos de trabajo por lotes (AWS SDK)

Proporcionar valores de filtro en su JSON de entrada

En el caso de los filtros con parámetros de marcadores de posición, como `$GENRE`, debe proporcionar los valores de los parámetros de un objeto `filterValues` en el JSON de entrada.

En el caso de un objeto `filterValues`, cada clave es un nombre de parámetro. Cada valor es el criterio que se pasa como parámetro. Rodee cada valor con comillas con caracteres de escape: `"filterValues":{"GENRES":"\"drama\""}`. Para varios valores, separe cada valor con una coma: `"filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}`

Ejemplo de JSON de entrada de trabajo de inferencia por lotes

El siguiente es un ejemplo de las primeras líneas de un archivo de entrada JSON para un trabajo de inferencia por lotes. El ejemplo incluye el objeto `filterValues`. La clave `GENRES` corresponde a un marcador de posición `$GENRES` en la expresión del filtro. El trabajo de este ejemplo usa la User-Personalization receta. Para las recetas RELATED_ITEMS, proporcione un itemId en lugar del userId. Para las recetas PERSONALIZED_RANKING, proporcione el userID y una itemList.

```
{"userId": "5","filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}}
{"userId": "3","filterValues":{"GENRES":"\"horror\",\"comedy\""}}
{"userId": "34","filterValues":{"GENRES":"\"drama\""}}
```

Para ver más ejemplos de datos de entrada de trabajos de inferencia por lotes según la receta, consulte Ejemplos de JSON de entrada y salida de trabajos de inferencia por lotes. Puede utilizar estos ejemplos como punto de partida y añadir el objeto `filterValues` del ejemplo anterior.

Ejemplo de JSON de entrada de trabajo de segmento por lotes

El siguiente es un ejemplo de las primeras líneas de un archivo de entrada JSON con valores de filtro para un trabajo de segmento por lotes. La clave `GENRES` corresponde a un marcador de posición `$GENRES` en la expresión del filtro.

```
{"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\""}}
```

Para ver más ejemplos de datos de entrada de trabajos de inferencia por lotes según la receta, consulte Ejemplos de JSON de entrada y salida de trabajos de segmentos por lotes. Puede utilizar estos ejemplos como punto de partida y añadir el objeto `filterValues` del ejemplo anterior.

Filtrado de flujos de trabajo por lotes (consola)

Para filtrar los flujos de trabajo por lotes con la consola de Amazon Personalize, debe crear un filtro y, a continuación, crear un trabajo de inferencia por lotes o un trabajo de segmento por lote y elegir el filtro. Para obtener instrucciones paso a paso, consulte Creación de un trabajo de inferencia por lotes (consola) y Creación de un trabajo de segmento por lotes (consola).

Filtrar flujos de trabajo por lotes (AWS SDK)

Para filtrar las recomendaciones de lotes con AWS los SDK, cree un filtro e incluya el `FilterArn` parámetro en la solicitud CreateBatchInferenceJob oCreateBatchSegmentJob.

En el siguiente código se muestra cómo crear un trabajo de inferencia por lotes con un filtro mediante AWS SDK para Python (Boto3). Recomendamos que utilice una ubicación diferente para sus datos de salida (ya sea una carpeta o un bucket de Amazon S3 diferente). Para obtener una explicación completa de todos los campos, consulte Creación de un trabajo de inferencia por lotes (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"}}
)
```

Registro de los eventos en tiempo real para influir en las recomendaciones

Un evento es una interacción entre un usuario y el catálogo. Puede ser una interacción con un elemento, como un usuario que compra un artículo o ve un vídeo, o se puede realizar una acción, como solicitar una tarjeta de crédito o registrarse en un programa de miembros.

Amazon Personalize puede hacer recomendaciones basadas únicamente en datos de eventos en tiempo real, únicamente en datos históricos de eventos o en una combinación de ambos. Registre los eventos en tiempo real a medida que los clientes interactúan con las recomendaciones. De esta forma, se construyen los datos de las interacciones y se mantienen actualizados. Además, se informa a Amazon Personalize sobre los intereses actuales del usuario, lo que puede mejorar la relevancia de las recomendaciones.

Si el caso de uso o la receta personalizada de su dominio admiten la personalización en tiempo real, Amazon Personalize utiliza eventos en tiempo real para actualizar y adaptar las recomendaciones según los intereses cambiantes del usuario.

La forma de registrar los eventos en tiempo real depende del tipo de datos de interacciones que importe:

- En el caso de las interacciones de elementos, se registran eventos en tiempo real con la operación de la API PutEvents. Amazon Personalize adjunta estos datos a los datos de interacciones de acción en el grupo de conjuntos de datos. Para obtener más información, consulte Registro de eventos de interacción de elemento en tiempo real.
- En el caso de las interacciones de acción, se registran eventos en tiempo real con la operación de la API PutActionInteractions. Amazon Personalize anexa estos datos al conjunto de datos de interacciones de acción en su grupo de conjuntos de datos. Solo las recetas PERSONALIZED_ACTIONS utilizan datos de interacciones de acción. Para obtener más información, consulte Registro de eventos de interacción de acciones en tiempo real.

Temas

- Influencia de los eventos en tiempo real en las recomendaciones
- Registro de eventos de interacción de elemento en tiempo real
- Registro de eventos de interacción de acciones en tiempo real

- Registro de eventos para usuarios anónimos
- Servicios de seguimiento de eventos de terceros
- Implementaciones de ejemplo

Influencia de los eventos en tiempo real en las recomendaciones

Si su receta admite la personalización en tiempo real, después de crear un recomendador o una campaña personalizada, Amazon Personalize utiliza los nuevos datos de eventos registrados para los elementos o acciones existentes segundos después de la importación. Los siguientes casos de uso y recetas permiten la personalización en tiempo real:

- Recomendaciones (caso de uso de ECOMMERCE)
- Mejores opciones (caso de uso de VIDEO_ON_DEMAND)
- Receta User-Personalization-v2
- Receta User-Personalization
- Receta Personalized-Ranking-v2
- Receta Personalized-Ranking
- Next-Best-Action receta

Si utiliza la receta Trending-Now, Amazon Personalize tiene en cuenta automáticamente los elementos de los datos de nuevos eventos en intervalos configurables. No tiene que crear una nueva versión de la solución. Para obtener más información, consulte Receta Trending-Now.

Si el elemento, la acción o el usuario del evento es nuevo, la forma en que Amazon Personalize utilice los datos dependerá de su caso de uso o receta. Para obtener más información, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Registro de eventos de interacción de elemento en tiempo real

Un evento de interacción de elementos es una interacción entre un usuario y un elemento del catálogo. Por ejemplo, un usuario que compra zapatos o ve una película.

Registre eventos de interacción de elementos en tiempo real al mostrar recomendaciones de elementos a los clientes. De esta forma, se construyen los datos de las interacciones y se mantienen

actualizados. Además, se informa a Amazon Personalize sobre los intereses actuales del usuario, lo que puede mejorar la relevancia de las recomendaciones.

Los eventos de interacción de elementos se registran con la operación de la API PutEvents. Amazon Personalize anexa los datos del evento al conjunto de datos de interacciones de elementos en su grupo de conjuntos de datos. Si registra dos eventos con exactamente la misma marca de tiempo y propiedades idénticas, Amazon Personalize solo conserva uno de ellos. Puede grabar los eventos de interacción entre elementos mediante AWS Amplify o AWS Command Line Interface ()AWS CLI. AWS SDKs

Si usa Apache Kafka, puede utilizar el conector Kafka para Amazon Personalize a fin de transmitir interacciones de elementos en tiempo real a Amazon Personalize. Para obtener más información, consulte Kafka Connector for Amazon Personalize en el repositorio de personalize-kafka-connectorGithub.

AWS Amplify incluye una JavaScript biblioteca para grabar eventos de interacción de elementos desde aplicaciones de cliente web y una biblioteca para grabar eventos en el código del servidor. Para obtener más información, consulte Documentación de Amplify.

Temas

- Requisitos para registrar eventos de interacción de elementos y entrenar un modelo
- Creación de un rastreador de eventos de interacción de elementos
- Registro de un solo evento de interacción de elementos
- Registro de varios eventos de interacción de elementos con datos de valores de eventos
- Registro de eventos de interacción de elemento con datos de impresiones
- Informes de atribución y métricas de eventos

Requisitos para registrar eventos de interacción de elementos y entrenar un modelo

Para registrar eventos de interacción de elementos, necesita lo siguiente:

- Un grupo de conjuntos de datos que incluya un conjunto de datos `Item interactions`, que puede estar vacío. Si ha realizado los procedimientos de la guía de Tutoriales de introducción, puede utilizar el mismo grupo de conjuntos de datos y el conjunto de datos que ha creado. Para obtener información sobre cómo crear un grupo de conjuntos de datos y un conjunto de datos,

consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.

- Un rastreador de eventos.
- Una llamada a la operación de la API PutEvents.
- Si utiliza una AWS Lambda función para llamar a la PutEvents operación, el rol de ejecución de la función debe tener permiso para realizar la `personalize:PutEvents` acción con el comodín `*` en el `Resource` elemento.

Puede empezar con un conjunto de datos de interacciones de elementos vacío y, cuando haya registrado suficientes datos, entrenar el modelo utilizando únicamente los nuevos eventos registrados. Para todos los casos de uso (grupos de conjuntos de datos de dominio) y recetas (grupos de recetas personalizadas), sus datos de interacciones deben tener lo siguiente antes del entrenamiento:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Al menos 25 usuarios únicos IDs con al menos dos interacciones entre elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

Creación de un rastreador de eventos de interacción de elementos

Antes de poder registrar eventos de interacción de elementos, debe crear un rastreador de eventos de este tipo. Un rastreador de eventos dirige los datos de eventos nuevos al conjunto de datos de interacciones de elementos de su grupo de conjuntos de datos.

Puede crear un rastreador de eventos con la consola de Amazon Personalize o la operación de la API CreateEventTracker. Debe pasar como parámetro el nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que contiene el conjunto de datos de interacciones de elementos de destino. Para obtener instrucciones sobre cómo crear un rastreador de eventos con la consola de Amazon Personalize, consulte Creación de un rastreador de eventos (consola).

Un rastreador de eventos incluye un identificador de seguimiento, que se transfiere como parámetro cuando se utiliza la PutEventsoperación. A continuación, Amazon Personalize anexa los datos del

nuevo evento al conjunto de datos de interacciones de elementos del grupo de conjuntos de datos que especifique en el rastreador de eventos.

> **Note**
>
> Solo puede crear un rastreador de eventos de interacciones de elementos para un grupo de conjuntos de datos.

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'])
```

Se muestran el ARN del rastreador de eventos y el ID de seguimiento, por ejemplo:

```
{
    "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
```

Se muestran el ARN del rastreador de eventos y el ID de seguimiento, por ejemplo:

```
{
```

```
    "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;
    }
```

Registro de un solo evento de interacción de elementos

Después de crear un conjunto de datos de interacciones de elementos y un rastreador de eventos para su grupo de conjuntos de datos, estará listo para registrar los eventos de interacción de elementos. En el ejemplo siguiente se muestra una operación `PutEvents` que pasa un evento de interacción de elementos. Se muestra el esquema correspondiente, junto con una fila de ejemplo del conjunto de datos de interacciones de elementos.

La aplicación genera un `sessionId` único cuando un usuario visita por primera vez su sitio web o utiliza su aplicación. Debe usar el mismo `sessionId` en todos los eventos de la sesión. Amazon Personalize usa `sessionId` para asociar eventos con el usuario antes de que inicie sesión (es anónimo). Para obtener más información, consulte Registro de eventos para usuarios anónimos.

La lista de eventos es una matriz de objetos Event. Se requiere `eventType` para cada evento. Si no tiene datos de tipo de evento, puede proporcionar un valor de marcador de posición para cumplir el requisito.

`trackingId` proviene del rastreador de eventos que creó en Creación de un rastreador de eventos de interacción de elementos. Los parámetros `userId`, `itemId` y `sentAt` se asignan a los campos USER_ID, ITEM_ID y TIMESTAMP de un conjunto de datos `Interactions` histórico correspondiente. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Columnas del conjunto de datos correspondientes

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE
Example data: user123, item-xyz, 1543631760, click
```

Ejemplo de código

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());
        }
}
```

Registro de varios eventos de interacción de elementos con datos de valores de eventos

Después de crear un conjunto de datos de interacciones de elementos y un rastreador de eventos para su grupo de conjuntos de datos, estará listo para registrar los eventos de interacción de elementos. En el ejemplo siguiente se muestra cómo registrar varios eventos de interacción de elementos con diferentes tipos y valores de eventos.

Al configurar una solución, si su conjunto de datos de interacciones de elementos incluye los campos EVENT_TYPE y EVENT_VALUE, puede establecer un valor específico como umbral para excluir registros del entrenamiento. Para obtener más información, consulte Elección de los datos de interacciones de elementos utilizados para el entrenamiento.

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**
>
> Las claves de propiedades utilizan nombres con la inicial en minúsculas que coinciden con los campos del esquema Interactions. Por ejemplo, si el campo “NUM_RATINGS” está definido en el esquema de interacciones, la clave de la propiedad debería ser `numRatings`.

Registro de eventos de interacción de elemento con datos de impresiones

Si usa la User-Personalization receta o agrega el campo IMPRESIONES a su esquema para un conjunto de datos de un grupo de conjuntos de datos de dominio, puede registrar los datos de impresiones en su PutEvents operación. Las impresiones son listas de elementos que estaban visibles para un usuario cuando interactuaba con un elemento concreto (por ejemplo, al hacer clic o verlo). Amazon Personalize utiliza los datos de impresiones para guiar la exploración, donde las recomendaciones incluyen elementos con menos interacciones, datos o relevancia. Para obtener información sobre las impresiones implícitas y explícitas que Amazon Personalize puede modelar, consulte Datos de impresiones.

> **Important**
>
> Si proporciona datos de impresiones implícitas y explícitas contradictorios en sus solicitudes de `PutEvents`. Amazon Personalize utiliza las impresiones explícitas de forma predeterminada.

Para registrar las recomendaciones de Amazon Personalize que muestra a sus usuarios como datos de impresiones, incluya `recommendationId` en su solicitud PutEvents y Amazon Personalize obtiene las impresiones implícitas en función de los datos de sus recomendaciones.

Para registrar manualmente los datos de impresiones de un evento, enumere las impresiones en el parámetro de entrada `impression` del comando PutEvents. En el siguiente ejemplo de código se muestra cómo incluir un `recommendationId` y un `impression` en una PutEvents operación con el

SDK para Python (Boto3) o el SDK para Java 2.x. Si incluyes ambos, Amazon Personalize utiliza las impresiones explícitas de forma predeterminada.

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

Utilice el siguiente método `putEvents` para registrar un evento con datos de impresiones y un recommendationId. Para el parámetro de impresiones, pasa la lista de ItemIDs como. 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());
    }
}
```

Informes de atribución y métricas de eventos

Para supervisar el tipo y la cantidad de eventos enviados a Amazon Personalize, utilice CloudWatch las métricas de Amazon. Para obtener más información, consulte Supervisión de Amazon Personalize con Amazon CloudWatch.

Para generar CloudWatch informes que muestren el impacto de las recomendaciones, cree una atribución métrica y registre las interacciones de los usuarios con recomendaciones en tiempo real. Para obtener información sobre cómo crear una atribución de métricas, consulte. Medición del impacto de las recomendaciones de Amazon Personalize

Para cada evento, incluya el ID de recomendación de las recomendaciones que mostró al usuario. O incluya el origen del evento, como un tercero. Importe estos datos para comparar diferentes campañas, recomendadores y terceros. Puede importar como máximo 100 fuentes de atribución de eventos.

- Si proporciona `recommendationId`, Amazon Personalize determina automáticamente la campaña o el recomendador de origen y lo identificará en los informes de una columna EVENT_ATTRIBUTION_SOURCE.
- Si proporciona ambos atributos, Amazon Personalize utilizará únicamente `eventAttributionSource`.
- Si no proporciona un origen, Amazon Personalize etiqueta el origen `SOURCE_NAME_UNDEFINED` en los informes.

El siguiente código muestra cómo proporcionar una `eventAttributionSource` para un evento en una PutEvents operación.

```
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)
```

El código siguiente muestra cómo proporcionar un `recommendationId` para un evento en una PutEvents operación.

```
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)
```

Registro de eventos de interacción de acciones en tiempo real

Un evento de interacción de acciones es una interacción entre un usuario y una acción. Por ejemplo, un usuario que se inscribe en un programa para miembros o que solicita una tarjeta de crédito.

Si usa una receta personalizada PERSONALIZED_ACTIONS, registre los eventos de interacción de acciones en tiempo real a medida que sus clientes interactúan con las recomendaciones de acción. De esta forma, se construyen los datos de las interacciones y se mantienen actualizados. También se informa a Amazon Personalize sobre los intereses actuales del usuario, lo que puede mejorar la relevancia de las recomendaciones. Solo las recetas personalizadas PERSONALIZED_ACTIONS utilizan datos de interacción de acciones.

Los eventos de interacción de acciones se registran con la operación de la API PutActionInteractions. Amazon Personalize anexa estos datos al conjunto de datos de interacciones de acción en su grupo de conjuntos de datos.

Un evento de interacción de acciones debe tener un atributo de tipo de evento, que puede ser uno de los siguientes:

- Realizado: registre los eventos como Realizado cuando un usuario realice una acción recomendada.
- No realizado: registre los eventos como No realizado cuando el usuario decida deliberadamente no realizar la acción después de verla. Por ejemplo, si elige No al mostrarle la acción. Los eventos de tipo No realizado pueden indicar que el cliente no está interesado en la acción.
- Visto: registre los eventos como Visto cuando muestre una acción al usuario antes de que tome la decisión de realizarla o no. Amazon Personalize utiliza los eventos Visto para obtener información sobre los intereses de los usuarios. Por ejemplo, si un usuario ve una acción pero no la realiza, puede que al usuario no le interese esta acción en el futuro.

Puede grabar eventos en tiempo real con AWS SDKs, o AWS Command Line Interface (AWS CLI). Si registra dos eventos con exactamente la misma marca de tiempo y propiedades idénticas, Amazon Personalize solo conserva uno de ellos.

Temas

- Requisitos para registrar eventos de interacción de acciones
- Búsqueda del ID de su rastreador de eventos de interacción de acciones
- Registro de un solo evento de interacción de acciones
- Registro de varios eventos de interacción de acciones

Requisitos para registrar eventos de interacción de acciones

Para registrar eventos de interacción de acciones en tiempo real, necesita lo siguiente:

- Un grupo de conjuntos de datos que incluya un `Action interactions dataset`, que puede estar vacío. Para obtener información sobre cómo crear un grupo de conjuntos de datos y un conjunto de datos, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.
- ID de su rastreador de eventos. Este identificador se especifica en la PutActionInteractions operación. Cuando crea un conjunto de datos de interacciones de acción, Amazon Personalize crea un rastreador de eventos de las interacciones de acción de forma automática. Para obtener más información, consulte Búsqueda del ID de su rastreador de eventos de interacción de acciones.
- Una llamada a la operación PutActionInteractions.

Búsqueda del ID de su rastreador de eventos de interacción de acciones

Cuando crea un conjunto de datos de interacciones de acción, Amazon Personalize crea un rastreador de eventos de las interacciones de acción de forma automática. El ID del rastreador se especifica en la operación de la PutActionInteractions API. Amazon Personalize lo usa para dirigir los nuevos datos al conjunto de datos de interacciones de acción en su grupo de conjuntos de datos.

Puede encontrar el ID del rastreador de eventos en la página de detalles del conjunto de datos de interacciones de acción en la consola de Amazon Personalize. Y puedes encontrar el ID llamando a la operación de la DescribeDataset API. El siguiente código Python imprime el ID de seguimiento de un conjunto de datos de interacciones de acción.

```
import boto3

personalize = boto3.client(service_name='personalize')
```

```
response = personalize.describe_dataset(
  datasetArn="Action interactions dataset ARN"
)

print(response['trackingId'])
```

Registro de un solo evento de interacción de acciones

Después de crear un conjunto de datos de interacciones de acción, ya puede registrar los eventos de interacción de acciones con la operación PutActionInteractions. En el siguiente código se muestra una operación `PutActionInteractions` que pasa un evento TAKEN. Puede registrar este evento cuando muestre a un usuario recomendaciones de Amazon Personalize y este realice alguna acción, como solicitar su tarjeta de crédito.

`actionInteractions`Es un conjunto de ActionInteraction objetos. El valor `trackingId` proviene del rastreador de eventos que Amazon Personalize generó al crear el usuario el conjunto de datos de interacciones de acción. Para obtener más información, consulte Búsqueda del ID de su rastreador de eventos de interacción de acciones.

La aplicación genera un `sessionId` único cuando un usuario visita por primera vez su sitio web o utiliza su aplicación. Debe usar el mismo `sessionId` en todos los eventos de la sesión. Amazon Personalize usa `sessionId` para asociar eventos con el usuario antes de que inicie sesión (es anónimo). Para obtener más información, consulte Registro de eventos para usuarios anónimos.

Los parámetros `userId`, `actionId` y `sentAt` se asignan a los campos USER_ID, ACTION_ID, EVENT_TYPE y TIMESTAMP del conjunto de datos de interacciones de acción.

Conjunto de datos de interacciones de acción correspondiente

```
USER_ID, ACTION_ID, TIMESTAMP, EVENT_TYPE
user123, action-xyz, 1543631760, TAKEN
```

Ejemplo de código

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'
  }]
)
```

Registro de varios eventos de interacción de acciones

El código siguiente muestra cómo registrar varios eventos de interacción de acciones para el mismo usuario con el mismo valor de sessionId.

Conjunto de datos de interacciones de acción correspondiente

```
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'
  }]
)
```

Registro de eventos para usuarios anónimos

Important

Si no registra al menos un evento con un usuario `sessionId` y `userId` para un usuario, Amazon Personalize no utilizará la actividad rastreada únicamente hasta `sessionId`

durante el entrenamiento. Y una vez finalizado el entrenamiento, las recomendaciones ya no se basarán en la actividad rastreada hasta `sessionId`. Esto creará un historial de eventos continuo para los ID de usuario antes y después de que inicien sesión.

Puede registrar eventos de interacción de elementos o de acciones para los usuarios antes de que creen una cuenta. Registre eventos para usuarios anónimos para crear un historial de eventos continuo con eventos anteriores y posteriores al inicio de sesión. Esto proporciona a Amazon Personalize más datos de interacciones sobre el usuario, lo que puede ayudar a generar recomendaciones más relevantes.

Para registrar eventos de usuarios anónimos (usuarios que no han iniciado sesión), especifique solo un `sessionId` para cada evento. La aplicación genera un `sessionId` único cuando un usuario visita por primera vez su sitio web o utiliza su aplicación. Debe usar el mismo `sessionId` en todos los eventos de la sesión. Amazon Personalize usa el `sessionId` para asociar eventos con el usuario antes de que inicie sesión.

Amazon Personalize no utiliza eventos de usuarios anónimos durante el entrenamiento hasta que los asocie a `userId`. Para obtener más información, consulte Creación de un historial de eventos continuo para usuarios anónimos.

Para proporcionar una personalización en tiempo real a los usuarios anónimos, especifique el SessionID como el UserID en su solicitud o solicitud. GetRecommendations GetActionRecommendations

- Para ver ejemplos de código que muestran cómo registrar eventos de interacción de elementos con la PutEvents operación y un SessionID y un UserID, consulte. Registro de un solo evento de interacción de elementos
- Para ver ejemplos de código que muestran cómo registrar eventos de interacción de acciones con la PutActionInteractions operación y un SessionID y un UserID, consulte. Registro de un solo evento de interacción de acciones

Creación de un historial de eventos continuo para usuarios anónimos

Para crear un historial de eventos para un usuario anónimo y hacer que Amazon Personalize utilice sus eventos durante el entrenamiento, registre al menos un evento con `sessionId` y `userId`. A continuación, puede registrar cualquier número de eventos para `userId`. Después de comenzar a proporcionar `userId`, `sessionId` puede cambiar. Durante el siguiente reentrenamiento completo,

Amazon Personalize asocia `userId` con el historial de usuarios anónimos rastreado al `sessionId` original.

Una vez finalizado el reentrenamiento, las recomendaciones se basarán en la actividad rastreada tanto de `sessionId` de los eventos anónimos como de cualquier evento rastreado hasta su `userId`.

> **Note**
>
> Si el usuario no ha creado una cuenta y quiere que Amazon Personalize utilice los datos durante el entrenamiento, puede utilizar `sessionId` como `userId` en los eventos. Sin embargo, si el usuario finalmente crea una cuenta, no podrá asociar los eventos de su navegación anónima con el nuevo `userId`.

Servicios de seguimiento de eventos de terceros

Las siguientes plataformas de datos de clientes (CDPs) pueden ayudarle a recopilar datos de eventos de su solicitud y enviarlos a Amazon Personalize.

- Amplitude: puede utilizar Amplitude para realizar un seguimiento de las acciones de los usuarios y así comprender su comportamiento. Para obtener información sobre el uso de Amplitude y Amazon Personalize, consulte la siguiente entrada del blog de la red de socios (APN) de AWS : Medición de la eficacia de la personalización con Amplitude y Amazon Personalize.
- Segment: puede usar Segment para enviar sus datos a Amazon Personalize. Para obtener más información sobre la integración de Segment con Amazon Personalize, consulte Destino de Amazon Personalize.

Implementaciones de ejemplo

Para ver un ejemplo de bloc de notas de Jupyter que muestra cómo usar Amazon Personalize para reaccionar ante el comportamiento en tiempo real de los usuarios que utilizan un rastreador de eventos y la PutEvents operación, consulte 2.view_Campaign_and_Interactions.ipynb en la carpeta getting_started del repositorio. amazon-personalize-samples GitHub

Para ver un ejemplo que muestre cómo transmitir eventos de usuarios que interactúan con las recomendaciones, consulte streaming_events en el repositorio de muestras de Amazon Personalize. GitHub

Para ver un ejemplo completo que contiene el código fuente y los archivos auxiliares para la implementación en tiempo real APIs que se encuentran entre los recursos de Amazon Personalize y las aplicaciones cliente, consulte Personalización APIs en tiempo real en el GitHub repositorio de AWS muestras. Este proyecto incluye cómo implementar lo siguiente:

- Contexto de usuario y recopilación de eventos de usuario
- Almacenamiento en caché de respuestas
- Recomendaciones de decoración basadas en los metadatos de los elementos
- Pruebas A/B
- Autenticación de API

Mantenimiento de la relevancia de las recomendaciones

Las recomendaciones relevantes pueden aumentar la captación de usuarios, la tasa de clics y la tasa de conversión para su aplicación a medida que crece su catálogo. Para mantener y mejorar la relevancia de las recomendaciones de Amazon Personalize para sus usuarios, mantenga actualizados sus datos y recursos personalizados. Esto permite a Amazon Personalize aprender del comportamiento más reciente de su usuario e incluir sus elementos más recientes en las recomendaciones.

Temas

- Mantener actualizados los conjuntos de datos
- Mantenimiento de los generadores de recomendaciones de dominios
- Mantenimiento de soluciones personalizadas

Mantener actualizados los conjuntos de datos

A medida que su catálogo crezca, actualice sus datos históricos con operaciones de importación de datos masivos o individuales. Para obtener más información acerca de la importación de datos históricos, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize. Para obtener información acerca de cómo los datos que importa después del entrenamiento de un modelo influyen en las recomendaciones, consulte Actualización de datos en conjuntos de datos después del entrenamiento.

Para casos de uso y recetas que proporcionan recomendaciones en tiempo real personalizadas, mantenga su conjunto de datos de interacciones de elementos actualizado en relación con el comportamiento de sus usuarios. Para ello, registre las interacciones de elementos con un rastreador de eventos y la operación de la API PutEvents. Amazon Personalize actualiza las recomendaciones en función de la actividad más reciente de su usuario a medida que interactúa con el catálogo. Para obtener información sobre la personalización en tiempo real, consulte Personalización en tiempo real. Para obtener más información sobre el registro de eventos en tiempo real, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.

Mantenimiento de los generadores de recomendaciones de dominios

Amazon Personalize vuelve a entrenar automáticamente los modelos que respaldan sus generadores de recomendaciones cada 7 días. Se trata de un reentrenamiento completo que crea modelos completamente nuevos basados en la totalidad de los datos de sus conjuntos de datos. Si modifica las columnas utilizadas en el entrenamiento, Amazon Personalize inicia automáticamente un reentrenamiento completo de los modelos que respaldan al recomendador.

- Para los casos de uso Mejores opciones y Recomendaciones, Amazon Personalize actualiza su generador de recomendaciones para considerar nuevos elementos para recomendaciones. Las actualizaciones automáticas no son un reentrenamiento completo en el que el modelo aprende del comportamiento de sus usuarios. En cambio, las actualizaciones automáticas permiten a Amazon Personalize incluir sus nuevos elementos en las recomendaciones antes del siguiente reentrenamiento completo del recomendador. Para obtener más información acerca de las actualizaciones automáticas, consulte Actualizaciones automáticas.
- SI usa el caso de uso Tendencia ahora, Amazon Personalize evalúa automáticamente los datos de sus interacciones cada dos hora e identifica los elementos que son tendencia. No es necesario que espere al reentrenamiento de su recomendador.

Mientras el reentrenamiento del recomendador esté en curso, podrá seguir recibiendo recomendaciones de él. Hasta que se complete el reentrenamiento, el recomendador utilizará la configuración y los modelos anteriores. Para realizar un seguimiento de las actualizaciones, puede ver la marca temporal de la última actualización del recomendador en la página Detalles del generador de recomendaciones en la consola de Amazon Personalize. O bien, puede ver los detalles de `latestRecommenderUpdate` de la operación DescribeRecommender.

Mantenimiento de soluciones personalizadas

De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático para crear una nueva versión de solución cada siete días. El entrenamiento continúa hasta que elimine la solución.

Al crear una solución, le recomendamos que utilice el entrenamiento automático para administrar la creación de versiones de solución. Esto facilita el mantenimiento de la solución. Elimina el entrenamiento manual necesario para que la solución aprenda a partir de los datos más recientes.

Sin un entrenamiento automático, debe crear de forma manual nuevas versiones de solución para que la solución aprenda a partir de los datos más recientes. Para obtener más información sobre la configuración del entrenamiento automático, consulte Configuración del entrenamiento automático.

La frecuencia de entrenamiento depende de los requisitos empresariales, de la receta que utilice y de la frecuencia con que importe los datos. Para todas las recetas, recomendamos el entrenamiento una vez a la semana como mínimo. Con el entrenamiento automático, esta es la frecuencia de entrenamiento predeterminada. Si agrega nuevos elementos o acciones con frecuencia, es recomendable que tenga una frecuencia de entrenamiento más alta, según la receta.

- Si utiliza User-Personalization-v2, User-Personalization o Next-Best-Action, la solución se actualiza automáticamente para tener en cuenta nuevos elementos o acciones para las recomendaciones. Las actualizaciones automáticas no son lo mismo que el entrenamiento automático. Una actualización automática no crea una versión de solución completamente nueva y el modelo no aprende a partir de los datos más recientes. Para mantener la solución, la frecuencia de entrenamiento debe seguir siendo de una vez a la semana como mínimo. Para obtener más información sobre las actualizaciones automáticas, incluidas las directrices y los requisitos adicionales, consulte Actualizaciones automáticas.
- Si usa Trending-Now, Amazon Personalize identifica automáticamente los elementos más populares en sus datos de interacciones durante un intervalo de tiempo configurable. Trending-Now puede recomendar elementos agregados desde el último entrenamiento mediante datos de interacciones masivos o en streaming. La frecuencia de entrenamiento debe seguir siendo de una vez a la semana como mínimo. Para obtener más información, consulte Receta Trending-Now.
- Si no utiliza una receta con actualizaciones automáticas ni la receta Trending-Now, Amazon Personalize tendrá en cuenta los elementos nuevos para las recomendaciones solo después del siguiente entrenamiento. Por ejemplo, si utiliza la receta Similar-Items y agrega nuevos elementos cada día, debe utilizar una frecuencia de entrenamiento diaria para que estos elementos aparezcan en las recomendaciones ese mismo día.

Actualización de datos en conjuntos de datos después del entrenamiento

A medida que el catálogo crezca, importe datos de entrenamiento adicionales a los conjuntos de datos. Esto contribuye a mantener y mejorar la relevancia de las recomendaciones de Amazon Personalize. Puede importar más datos con operaciones de importación de datos masivos o individuales.

- Con las importaciones individuales, Amazon Personalize agrega los nuevos registros al conjunto de datos. Para actualizar un elemento, usuario o acción individual, puede importar un registro con el mismo ID pero con los atributos modificados. Puede importar hasta 10 registros por operación de importación individual.

  Para obtener más información sobre la importación de registros de forma individual, consulte Importación de registros individuales a un conjunto de datos de Amazon Personalize. Para obtener información sobre el registro de eventos en tiempo real, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.
- Con las importaciones masivas, puede agregar o reemplazar datos masivos mediante la creación de otro trabajo de importación. De forma predeterminada, un trabajo de importación de conjuntos de datos reemplaza cualquier dato existente en el conjunto de datos que haya importado de manera masiva. En su lugar, puede anexar los nuevos registros a los datos existentes cambiando el modo de importación del trabajo.

  Para anexar datos a un conjunto de datos de interacciones de elementos o de interacciones de acción con un trabajo de importación de conjuntos de datos, debe tener al menos 1000 registros de interacciones de elementos o de interacciones de acción nuevos. A los 20 minutos de completar una importación masiva, Amazon Personalize actualiza todos los filtros que haya creado en el grupo de conjuntos de datos con sus nuevos datos en masa. Esta actualización permite a Amazon Personalize utilizar los datos más recientes al filtrar las recomendaciones para sus usuarios.

Después de crear un conjunto de datos de elementos o usuarios, puede reemplazar su esquema por uno nuevo o uno ya existente. Puede reemplazar el esquema de un conjunto de datos si su estructura de datos cambió después de crear el conjunto de datos. Por ejemplo, puede tener una nueva columna de metadatos de elementos que desee que Amazon Personalize tenga en cuenta durante el entrenamiento. O tal vez quiera añadir una columna de datos para utilizarla únicamente al

filtrar las recomendaciones. Para obtener más información, consulte Reemplazo del esquema de un conjunto de datos para agregar nuevas columnas.

Después de crear un recomendador o una versión de solución personalizada, la forma en que los nuevos datos influyen en las recomendaciones depende del tipo, el método de importación y el caso de uso de dominio o la receta personalizada que utilice. En las siguientes secciones se explica cómo los datos nuevos influyen en las recomendaciones en tiempo real y por lotes antes del siguiente entrenamiento.

Temas

- Influencia de los nuevos datos en las recomendaciones en tiempo real
- Cómo influyen los datos nuevos en las recomendaciones por lotes (recursos personalizados)

Influencia de los nuevos datos en las recomendaciones en tiempo real

Después de crear un recomendador o una versión de solución personalizada, la forma en que los nuevos datos influyen en las recomendaciones depende del tipo de los datos, el método de importación y el caso de uso de dominio o la receta personalizada que utilice. En las siguientes secciones se explica cómo los datos nuevos influyen en las recomendaciones en tiempo real antes del siguiente entrenamiento.

El entrenamiento puede ser un entrenamiento automático semanal de un recomendador o la creación de una versión de solución automática o manual. En el entrenamiento manual con User-Personalization, omita `trainingMode` para usar el modo de entrenamiento `FULL` predeterminado.

Temas

- Nuevas interacciones
- Elementos nuevos
- Nuevos usuarios
- Nuevas acciones

Nuevas interacciones

Las interacciones nuevas son interacciones de elemento o de acción que se importan después del último entrenamiento. Tanto para los datos en tiempo real como masivos, si las interacciones

implican un nuevo elemento o acción, Amazon Personalize podría tenerlo en cuenta para las recomendaciones si el caso de uso o la receta incluyen exploración. Para obtener más información, consulte Elementos nuevos o Nuevas acciones.

Eventos en tiempo real

Para los casos de uso y las recetas que incluyen la personalización en tiempo real, Amazon Personalize utiliza inmediatamente interacciones en tiempo real entre un usuario y elementos o acciones existentes en el último entrenamiento. Al generar recomendaciones para el usuario en el evento, Amazon Personalize utiliza estas interacciones en tiempo real. Para obtener más información sobre la personalización en tiempo real, consulte Personalización en tiempo real.

Para cualquier caso de uso de dominio y receta personalizada que no incluya la personalización en tiempo real, como la recomendación de elementos similares, el modelo aprende a partir de los datos de las interacciones en tiempo real solo después del entrenamiento.

Interacciones masivas

En el caso de las interacciones masivas, tanto para los trabajos de importación de conjuntos de datos incrementales como completos, el modelo aprende de los datos de las interacciones de acciones o de elementos masivas solo después del siguiente entrenamiento. Los datos masivos no se utilizan para actualizar las recomendaciones para una personalización en tiempo real.

Para obtener más información sobre cómo importar más datos masivos, consulte Importación de datos masivos en Amazon Personalize con un trabajo de importación de conjuntos de datos.

Elementos nuevos

Los elementos nuevos son aquellos que se importan después del último entrenamiento. Pueden provenir de datos de interacciones o de metadatos de elementos en un conjunto de datos de elementos.

Los elementos nuevos se tienen en cuenta para las recomendaciones de la siguiente manera:

- Para los casos de dominio Mejores opciones y Recomendado para usted o las recetas User-Personalization-v2, User-Personalization o Next-Best-Action, Amazon Personalize actualiza automáticamente el modelo cada dos horas. Tras cada actualización, Amazon Personalize tendrá en cuenta nuevos elementos para las recomendaciones como parte de la exploración. Al considerar el nuevo elemento, Amazon Personalize tiene en cuenta todos los metadatos para

este. Sin embargo, estos datos tendrán un mayor efecto en las recomendaciones solo después de registrar las interacciones del elemento y entrenar un modelo nuevo. Para obtener información sobre las actualizaciones, consulte Actualizaciones automáticas.

- SI usa el caso de uso Tendencia ahora, Amazon Personalize evalúa automáticamente los datos de sus interacciones cada dos hora e identifica los elementos que son tendencia. No es necesario que espere al entrenamiento del recomendador. Si utiliza la receta Trending-Now, Amazon Personalize tiene en cuenta automáticamente todos los elementos nuevos en intervalos configurables sin entrenamiento. Para obtener más información sobre la configuración de los intervalos, consulte Receta Trending-Now.
- Si no utiliza la receta Trending-Now o el caso de uso o receta no admiten las actualizaciones automáticas, Amazon Personalize considerará los elementos nuevos solo después del siguiente entrenamiento.

Nuevos usuarios

Los usuarios nuevos son aquellos que se importan después del último entrenamiento. Pueden provenir de datos de interacciones o de metadatos de usuarios en un conjunto de datos de usuarios. Para los usuarios nuevos y anónimos (usuarios sin un userId), puede registrar los eventos del usuario con un `sessionId` y Amazon Personalize asociará los eventos al usuario antes de que inicie sesión. Para obtener más información, consulte Registro de eventos para usuarios anónimos.

Amazon Personalize genera recomendaciones para los nuevos usuarios de la siguiente manera:

- Si utiliza el caso de uso del dominio Tendencia ahora o la receta personalizada Trending-Now, los nuevos usuarios recibirán inmediatamente recomendaciones sobre los elementos que son tendencia. Si utiliza la receta Popularity-Count, los nuevos usuarios recibirán inmediatamente recomendaciones sobre los elementos con más interacciones.
- En el caso de recetas o casos de uso que ofrecen recomendaciones personalizadas a los usuarios, las recomendaciones para los nuevos usuarios se basan en los historiales de interacciones iniciales de los usuarios actuales. Es más probable que se recomienden a los nuevos usuarios los primeros elementos o acciones con los que hayan interactuado los usuarios existentes. En el caso de las recetas User-Personalization o Personalized-Ranking, si establece `recency_mask` en `true`, las recomendaciones también incluyen elementos basados en las últimas tendencias de popularidad en sus datos de interacciones.

Lo siguiente puede aumentar la relevancia de las recomendaciones para los nuevos usuarios:

- Datos de interacciones: la forma principal de mejorar la relevancia de las recomendaciones para un nuevo usuario es importar los datos de sus interacciones con sus elementos. Para obtener información sobre cómo los datos de las nuevas interacciones influyen en las recomendaciones, consulte Nuevas interacciones.
- Metadatos de usuario: la importación de metadatos de usuario, como GENDER o MEMBERSHIP_STATUS, puede mejorar las recomendaciones. Para que los metadatos influyan en las recomendaciones, debes esperar a que el recomendador de dominios complete su reentrenamiento automático semanal. O bien, tiene que crear manualmente una nueva versión de la solución.
- Metadatos contextuales: si su caso de uso o receta admite metadatos contextuales y su conjunto de datos de interacciones de elementos tiene campos de metadatos para datos contextuales, puede proporcionar el contexto del usuario en su solicitud de recomendaciones. Esto no requiere un reentrenamiento. Para obtener más información, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales.

Nuevas acciones

Las acciones nuevas son aquellas que se importan desde el último entrenamiento. Pueden provenir de los datos de interacción de acciones o de las acciones de un conjunto de datos de acciones.

Con la receta Next-Best-Action, Amazon Personalize actualiza automáticamente una versión de la solución cada dos horas. Después de cada actualización, Amazon Personalize considera nuevas acciones para recomendarlas como parte de la exploración. Al considerar la nueva acción, Amazon Personalize tiene en cuenta todos los metadatos para esta. Sin embargo, estos datos tendrán un mayor efecto en las recomendaciones solo después de registrar las interacciones de la acción y realizar un reentrenamiento completo. Para obtener información sobre las actualizaciones, consulte Actualizaciones automáticas

Cómo influyen los datos nuevos en las recomendaciones por lotes (recursos personalizados)

Después de crear una versión de solución personalizada, la forma en que los nuevos datos influyen en las recomendaciones depende del tipo de los datos, el método de importación y la receta personalizada que utilice.

Para segmentos de usuario, Amazon Personalize genera segmentos utilizando únicamente los datos masivos presentes en el último entrenamiento de la versión completa de la solución. Además, Amazon Personalize solo utiliza datos masivos que haya importado usted con el modo de importación FULL (para reemplazar los datos existentes). Para obtener más información sobre los segmentos de usuario, consulte Obtención de segmentos de usuarios por lotes con recursos personalizados.

Al generar recomendaciones de elementos por lotes, Amazon Personalize tiene en cuenta todos los datos masivos presentes en el momento de la creación de la última versión de solución. Estos datos se pueden importar con un modo de importación COMPLETO o INCREMENTAL. Para que los registros masivos más recientes influyan en las recomendaciones por lotes, debe crear una nueva versión de la solución y, a continuación, crear el trabajo de inferencia por lotes.

En las secciones siguientes se explica la forma en que las importaciones individuales influyen en las recomendaciones de elementos por lotes.

Temas

- Nuevas interacciones
- Nuevos usuarios
- Elementos nuevos

Nuevas interacciones

Si utiliza una receta USER_PERSONALIZATION o PERSONALIZED_RANKING, Amazon Personalize tiene en cuenta nuevos datos de interacciones de elemento con los elementos y los usuarios existentes unos 15 minutos a partir de la importación de los datos. Estos elementos y usuarios deben haber estado presentes en el último entrenamiento. Para asegurarse de que se tienen en cuenta los eventos, recomendamos que espere al menos 15 minutos antes de iniciar un trabajo de inferencia por lotes. Para todas las demás recetas, y para los eventos con nuevos elementos o usuarios, debe crear una nueva versión de solución para los eventos transmitidos para influir en las recomendaciones por lotes.

Nuevos usuarios

Para los usuarios sin datos de interacciones, inicialmente las recomendaciones solo se refieren a elementos populares. Si utiliza la receta USER_PERSONALIZATION o PERSONALIZED_RANKING, y registra eventos para el usuario, las recomendaciones podrían ser más relevantes unos 15 minutos

después de la importación sin necesidad de reentrenamiento. Para asegurarse de que se tienen en cuenta los eventos, recomendamos que espere al menos 15 minutos antes de iniciar un trabajo de inferencia por lotes. Para todas las demás recetas, debe crear una nueva versión de la solución para los eventos transmitidos para influir en las recomendaciones por lotes para los usuarios sin datos de interacciones.

Elementos nuevos

Con User-Personalization-v2 y User-Personalization, al crear un trabajo de inferencia por lotes y especificar la última versión de solución completamente entrenada, Amazon Personalize actualiza automáticamente la versión de solución para incluir nuevos elementos en las recomendaciones con la exploración. Si no especifica la versión de la solución más reciente, no se produce ninguna actualización. Para cualquier otra receta, debe crear una nueva versión de la solución para que los nuevos elementos aparezcan en las recomendaciones por lotes. Para obtener más información acerca de la exploración, consulte Exploration (Exploración).

Reemplazo del esquema de un conjunto de datos para agregar nuevas columnas

Después de crear un conjunto de datos de elementos o usuarios, puede reemplazar su esquema por uno nuevo o uno ya existente. Puede reemplazar el esquema de un conjunto de datos si su estructura de datos cambió después de crear el conjunto de datos. Por ejemplo, puede tener una nueva columna de metadatos de elementos que desee que Amazon Personalize tenga en cuenta durante el entrenamiento. O tal vez quiera añadir una columna de datos para utilizarla únicamente al filtrar las recomendaciones.

Al reemplazar el esquema de un conjunto de datos, debe conservar todos los campos del esquema anterior y no puede cambiar sus tipos de datos o atributos. Tras reemplazar el esquema de un conjunto de datos, Amazon Personalize excluye automáticamente del entrenamiento cualquier columna nueva para cualquier recomendador o solución personalizada existente. Para obtener más pautas y requisitos, consulte Directrices y requisitos.

Puede reemplazar el esquema de un conjunto de datos por la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) y AWS los SDK.

Temas

- Directrices y requisitos
- Sustitución del esquema de un conjunto de datos (consola)
- Sustituir el esquema de un conjunto de datos (AWS CLI)
- Reemplazar el esquema de un conjunto de datos (AWS SDK)

Directrices y requisitos

Antes de reemplazar el esquema para un conjunto de datos, asegúrese de que conoce las siguientes pautas y requisitos:

- No se puede reemplazar el esquema de un conjunto de datos de interacciones de elementos, un conjunto de datos de interacciones de acción o un conjunto de datos de acciones.
- Puede añadir nuevos campos al esquema de reemplazo, pero debe conservar todos los campos del esquema anterior. Y no puede cambiar sus tipos de datos o atributos. Por ejemplo, si el esquema anterior incluye un campo `MEMBERSHIP_STATUS` para cadenas de datos categóricos,

el nuevo esquema que utilice debe incluir un campo `MEMBERSHIP_STATUS` con estos atributos y tipos de datos.

- Si el esquema actual tiene un campo al que desea cambiar el nombre, o si desea cambiar sus tipos de datos o atributos, puede agregar un campo nuevo con un nombre nuevo y tipos o atributos modificados. A continuación, incluya el campo nuevo en el entrenamiento y excluya el campo anterior. Todos los campos nuevos deben admitir datos `null`. Si el campo anterior no admitía datos de anulación, al importar datos, puede usar datos de marcador de posición para asegurarse de que la importación coincide con el esquema. Para obtener información sobre la configuración de las columnas utilizadas por un recomendador, consulte Actualización de un recomendador. Para obtener información sobre la configuración de las columnas utilizadas por una solución, consulte Configuración de las columnas que se utilizan en el entrenamiento.
- Todos los campos nuevos deben admitir datos `null`. Para obtener más información acerca de cómo añadir un tipo nulo a un campo, consulte Tipos de datos de esquema.
- Tras reemplazar el esquema de un conjunto de datos, Amazon Personalize excluye automáticamente del entrenamiento cualquier columna nueva para cualquier recomendador o solución personalizada existente. El uso del conjunto de datos modificado implica las siguientes acciones:
  - Para usar cualquier columna nueva en el entrenamiento, importe los datos que se alineen con el nuevo esquema. A continuación, actualice las columnas recomendadas para que usen columnas nuevas o cree una nueva solución personalizada y configure las columnas que usará durante el entrenamiento.

    Para obtener información sobre la actualización de las columnas utilizadas por un recomendador, consulte Actualización de un recomendador. Para obtener información sobre la configuración de las columnas utilizadas por una solución, consulte Configuración de las columnas que se utilizan en el entrenamiento.
  - Para usar cualquier columna solo al filtrar, importe los datos que se alineen con el nuevo esquema, cree un filtro que utilice los nuevos datos y aplique el filtro a las solicitudes de recomendación. No necesita actualizar ningún recomendador ni crear o actualizar ningún recurso personalizado.

Sustitución del esquema de un conjunto de datos (consola)

Para sustituir el esquema de un conjunto de datos por la consola de Amazon Personalize, debe elegir el conjunto de datos que desee modificar y sustituirlo por un esquema nuevo o utilizar uno existente.

Para reemplazar el esquema de un conjunto de datos

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
3. En el panel de navegación, elija Conjuntos de datos y elija el botón de opción correspondiente al conjunto de datos que desea modificar.
4. Elija Acciones y, a continuación, elija Reemplazar esquema.
5. En Detalles del esquema, elija reemplazarlo por un esquema nuevo o por uno creado anteriormente.
6. Especifique el nuevo esquema que se va a utilizar. Si ha optado por:
   - Sustituir por un esquema nuevo, asígnele un nombre y, en la Definición de esquema, realice los cambios en el JSON del esquema.
   - Usar un esquema creado anteriormente, en Esquema creado anteriormente, elija el esquema que quiere usar. Solo se muestran los esquemas que reúnen las condiciones. Para obtener información sobre los requisitos de esquema, consulte Directrices y requisitos.
7. Elija Reemplazar. Cuando el conjunto de datos esté activo, puede empezar a importar datos que se alineen con el nuevo esquema. Para obtener más información, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.

Sustituir el esquema de un conjunto de datos (AWS CLI)

Para reemplazar el esquema de un conjunto de datos por el AWS CLI, usa el `update-dataset` comando que especifica el nombre de recurso de Amazon (ARN) del conjunto de datos que se va a actualizar y el ARN del nuevo esquema que se va a usar. No se puede actualizar el esquema de un conjunto de datos de interacciones de elementos, un conjunto de datos de interacciones de acciones o un conjunto de datos de acciones.

El siguiente código muestra cómo actualizar el esquema de un conjunto de datos con AWS CLI. Para reemplazar el esquema de un conjunto de datos por uno nuevo, use primero el comando `create-schema`. A continuación, utilice el siguiente código para reemplazar el esquema actual por el nuevo. Para obtener información sobre cómo crear un esquema con el AWS CLI, consulte. Creación de un conjunto de datos y un esquema (AWS CLI) Para obtener información sobre los conjuntos de datos y los requisitos del esquema, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

```
aws personalize update-dataset \
--dataset-arn Dataset ARN \
--schema-arn New schema ARN
```

Cuando el conjunto de datos esté activo, puede empezar a importar datos que se alineen con el nuevo esquema. Para obtener más información, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize. Para obtener información sobre la última actualización del conjunto de datos, puede utilizar la operación DescribeDataset.

Reemplazar el esquema de un conjunto de datos (AWS SDK)

Para reemplazar el esquema de un conjunto de datos por los AWS SDK, se utiliza la operación `UpdateDataset` API. Especifique el nombre de recurso de Amazon (ARN) del conjunto de datos que desea actualizar y el esquema nuevo que desea utilizar. No se puede actualizar el esquema de un conjunto de datos de interacciones de elementos, un conjunto de datos de interacciones de acciones o un conjunto de datos de acciones.

En el siguiente código se muestra cómo reemplazar el esquema de un conjunto de datos por el AWS SDK para Python (Boto3). Para reemplazar el esquema de un conjunto de datos por uno nuevo, use primero la operación CreateSchema. A continuación, utilice el siguiente código para reemplazar el esquema actual por el nuevo. Para obtener información sobre cómo crear un esquema con los AWS SDK, consulteCreación de un conjunto de datos y un esquema (AWS SDKs). Para obtener información sobre los requisitos de los conjuntos de datos y el esquema, consulte Creación de archivos JSON de esquema para los esquemas de 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)
```

Cuando el conjunto de datos esté activo, puede empezar a importar datos que se alineen con el nuevo esquema. Para obtener más información, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize. Para obtener información sobre la última actualización del conjunto de datos, puede utilizar la operación DescribeDataset.

Exportación de los datos de entrenamiento de un conjunto de datos a Amazon S3

Tras importar los datos a un conjunto de datos de Amazon Personalize, puede exportar los datos a un bucket de Amazon S3. Puede exportar datos para verificar e inspeccionar los datos que Amazon Personalize utiliza para generar recomendaciones, ver los eventos de interacción de elementos que registró anteriormente en tiempo real o realizar análisis de sus datos sin conexión a Internet.

Puede elegir exportar solo los datos que haya importado de manera masiva (importados mediante un trabajo de importación de conjuntos de datos de Amazon Personalize), solo los datos que haya importado individualmente (registros importados mediante la consola o las operaciones `PutEvents`, `PutUsers` o `PutItems`), o ambos.

> **Note**
>
> No se pueden exportar datos en un conjunto de datos de interacciones de acción ni en un conjunto de datos de acciones.

Para los registros que coinciden exactamente con todos los campos, Amazon Personalize exporta solo un registro. Si dos registros tienen el mismo ID pero uno o más campos son diferentes, Amazon Personalize incluye o elimina los registros en función de los datos que decida exportar:

- Si exporta datos tanto masivos como incrementales, Amazon Personalize solo exporta los elementos más recientes con el mismo ID (en las exportaciones de conjuntos de datos de elementos) y solo los usuarios con el mismo ID (en las exportaciones de conjuntos de datos de usuarios). Para los conjuntos de datos de interacciones de elementos, Amazon Personalize exporta todos los datos de interacciones de elementos.
- Si solo exporta datos incrementales, Amazon Personalize exporta todos los datos de elementos, usuarios o interacciones de elementos que haya importado individualmente, incluidos los elementos o usuarios con los mismos ID. Solo se excluyen los registros que coincidan exactamente para todos los campos.
- Si solo exporta datos masivos, Amazon Personalize incluye todos los datos de elementos, usuarios o interacciones de elementos que haya importado de manera masiva, incluidos los elementos o

usuarios con los mismos ID. Solo se excluyen los registros que coincidan exactamente para todos los campos.

Para exportar un conjunto de datos, cree un trabajo de exportación de conjunto de datos. Un trabajo de exportación de conjuntos de datos es una herramienta de exportación de registros que envía los registros de un conjunto de datos a uno o más archivos CSV de un bucket de Amazon S3. El archivo CSV de salida incluye una fila de encabezado con nombres de columnas que coinciden con los campos del esquema del conjunto de datos.

Temas

- Requisitos de permisos para trabajos de exportación de conjuntos de datos
- Creación de un trabajo de exportación de conjuntos de datos en Amazon Personalize

Requisitos de permisos para trabajos de exportación de conjuntos de datos

Para exportar un conjunto de datos, Amazon Personalize necesita permiso para añadir archivos al bucket de Amazon S3. Para conceder permisos, asocie una nueva AWS Identity and Access Management (política de IAM) a su rol de servicio de Amazon Personalize que concede el permiso de rol para usar las acciones `PutObject` y `ListBucket` de su bucket, y asocie una política de bucket a su bucket de Amazon S3 de salida que conceda el permiso de principio de Amazon Personalize para usar las acciones `PutObject` y `ListBucket`.

Si utilizas AWS Key Management Service (AWS KMS) para el cifrado, debes conceder permiso a Amazon Personalize y a tu rol de servicio Amazon Personalize IAM para usar tu clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Política de roles de servicio para exportar un conjunto de datos

El siguiente ejemplo de política concede a su rol de servicio de Amazon Personalize permiso para usar las acciones `PutObject` y `ListBucket`. Sustituya `amzn-s3-demo-bucket` por el nombre del bucket de salida. Para obtener información acerca de la asociación de políticas a un rol de servicio de IAM, consulte Asociación de una política de Amazon S3 a su rol de servicio de 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/*"
            ]
        }
    ]
}
```

Política de buckets de Amazon S3 para exportar un conjunto de datos

El siguiente ejemplo de política concede permiso a Amazon Personalize para usar las acciones `PutObject` y `ListBucket` en un bucket de Amazon S3. Reemplace `amzn-s3-demo-bucket` con el nombre de su bucket. Para obtener más información sobre cómo agregar una política de bucket de Amazon S3, consulte Agregar una política de bucket mediante la consola de Amazon S3 en la Guía del usuario de 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/*"
            ]
        }
    ]
}
```

Creación de un trabajo de exportación de conjuntos de datos en Amazon Personalize

Puede crear un trabajo de exportación de conjuntos de datos con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS los SDK.

Creación de un trabajo de exportación de conjunto de datos (consola)

Tras importar sus datos en un conjunto de datos y crear un bucket de Amazon S3 de salida, puede exportar los datos en el bucket para su análisis. Para exportar un conjunto de datos mediante la consola Amazon Personalize, debe crear un trabajo de exportación de conjuntos de datos. Para obtener información sobre la creación de un bucket de Amazon S3, consulte Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

Antes de exportar un conjunto de datos, asegúrese de que su rol de servicio de Amazon Personalize pueda acceder y escribir en su bucket de Amazon S3 de salida. Consulte Requisitos de permisos para trabajos de exportación de conjuntos de datos.

Para crear un trabajo de exportación de conjunto de datos (consola)

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/home.
2. En el panel de navegación, elija Grupos de conjuntos de datos.
3. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
4. Elija Conjuntos de datos en el panel de navegación.
5. Elija el conjunto de datos que desea exportar en un bucket de Amazon S3.
6. En Trabajos de exportación de conjuntos de datos, elija Crear trabajo de exportación de conjuntos de datos.
7. En Detalles del trabajo de exportación de conjuntos de datos, para Nombre del trabajo de exportación de conjuntos de datos, especifique un nombre para el trabajo de exportación.
8. Para el Rol de servicio de IAM, elija el rol de servicio de Amazon Personalize que creó en Creación de un rol de IAM para Amazon Personalize.

9. Para la Ruta de salida de datos de Amazon S3, especifique el bucket de Amazon S3 de destino. Utilice la siguiente sintaxis:

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

10. Si lo utiliza AWS KMS para el cifrado, para el ARN de la clave de KMS, introduzca el nombre de recurso de Amazon (ARN) de la clave. AWS KMS
11. En el Tipo de datos de exportación, elija el tipo de datos que desee exportar en función de cómo los importó originalmente.

    - Seleccione Masivo para exportar solo los datos que haya importado de forma masiva mediante un trabajo de importación de conjuntos de datos.
    - Elija Incremental para exportar solo los datos que haya importado individualmente mediante la consola o las operaciones `PutEvents`, `PutUsers` o `PutItems`.
    - Elija Ambos para exportar todos los datos del conjunto de datos.
12. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.
13. Seleccione Crear trabajo de exportación de conjuntos de datos.

    En la página Información general del conjunto de datos, en Trabajos de exportación de conjuntos de datos, el trabajo aparece con un Estado de trabajo de exportación. El trabajo de exportación del conjunto de datos se completa cuando el estado cambia a ACTIVE. A continuación, puede descargar los datos desde el bucket de Amazon S3 de salida. Para obtener información sobre cómo descargar objeto de un bucket de Amazon S3, consulte Descarga de un objeto en la Guía del usuario de Amazon Simple Storage Service.

Crear un trabajo de exportación de conjuntos de datos (AWS CLI)

Tras importar los datos en el conjunto de datos y crear un bucket de Amazon S3 de salida, puede exportar el conjunto de datos al bucket para su análisis. Para exportar un conjunto de datos mediante el AWS CLI, cree un trabajo de exportación de conjuntos de datos mediante el `create-dataset-export-job` AWS CLI comando. Para obtener información sobre la creación de un bucket de Amazon S3, consulte Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

Antes de exportar un conjunto de datos, asegúrese de que el rol de servicio de Amazon Personalize pueda acceder y escribir en su bucket de Amazon S3 de salida. Consulte Requisitos de permisos para trabajos de exportación de conjuntos de datos.

A continuación, se muestra un ejemplo del comando `create-dataset-export-job` AWS CLI . Asigne un nombre al trabajo, sustituya `dataset arn` por el Nombre de recurso de Amazon (ARN) del conjunto de datos que desea exportar y sustituya `role ARN` por el ARN del rol de servicio de Amazon Personalize que creó en Creación de un rol de IAM para Amazon Personalize. En `s3DataDestination`, para el `kmsKeyArn`, si lo desea, proporcione el ARN de su AWS KMS clave y, para el, `path` proporcione la ruta al bucket de Amazon S3 de salida.

Para `ingestion-mode`, especifique los datos que se exportarán de las siguientes opciones:

- Especifique `BULK` para exportar solo los datos que haya importado de forma masiva mediante un trabajo de importación de conjuntos de datos.
- Especifique `PUT` que solo se exporten los datos que haya importado individualmente mediante la consola o las `PutItems` operaciones `PutEvents` PutUsers, o.
- Especifique `ALL` para exportar todos los datos del conjunto de datos.

Para obtener más información, consulte 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
```

Se muestra el ARN del trabajo de exportación del conjunto de datos.

```
{
  "datasetExportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-export-job/
DatasetExportJobName"
}
```

Utilice la operación `DescribeDatasetExportJob` para comprobar el estado.

```
aws personalize describe-dataset-export-job \
```

```
  --dataset-export-job-arn dataset export job ARN
```

Crear un trabajo de exportación de conjuntos de datos (AWS SDK)

Tras importar los datos en el conjunto de datos y crear un bucket de Amazon S3 de salida, puede exportar el conjunto de datos al bucket para su análisis. Para exportar un conjunto de datos mediante los AWS SDK, cree un trabajo de exportación de conjuntos de datos mediante la CreateDatasetExportJob operación. Para obtener información sobre la creación de un bucket de Amazon S3, consulte Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

El código siguiente muestra cómo crear un trabajo de exportación de conjuntos de datos mediante el SDK AWS SDK para Python (Boto3) o el SDK para Java 2.x.

Antes de exportar un conjunto de datos, asegúrese de que el rol de servicio de Amazon Personalize pueda acceder y escribir en su bucket de Amazon S3 de salida. Consulte Requisitos de permisos para trabajos de exportación de conjuntos de datos.

SDK for Python (Boto3)

Utilice el siguiente `create_dataset_export_job` para exportar los datos de un conjunto de datos en un bucket de Amazon S3. Asigne un nombre al trabajo, sustituya `dataset arn` por el nombre de recurso de Amazon (ARN) del conjunto de datos que desea exportar y sustituya `role ARN` por el ARN del rol de servicio de Amazon Personalize que creó en Creación de un rol de IAM para Amazon Personalize. En `s3DataDestination`, para el `kmsKeyArn`, si lo desea, proporcione el ARN de su AWS KMS clave y, para el, `path` proporcione la ruta al bucket de Amazon S3 de salida.

Para `ingestionMode`, especifique los datos que se exportarán de las siguientes opciones:

- Especifique `BULK` para exportar solo los datos que haya importado de forma masiva mediante un trabajo de importación de conjuntos de datos.
- Especifique `PUT` que solo se exporten los datos que haya importado individualmente mediante la consola o las `PutItems` operaciones `PutEvents` PutUsers, o.
- Especifique `ALL` para exportar todos los datos del conjunto de datos.

```
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

Utilice el siguiente método `createDatasetExportJob` para crear un trabajo de exportación de conjunto de datos. Pase los siguientes parámetros como parámetros: a PersonalizeClient, el nombre del trabajo de exportación, el ARN del conjunto de datos que desea exportar, el modo de ingesta, la ruta del bucket Amazon S3 de salida y el ARN de la clave. AWS KMS

El `ingestionMode` puede ser una de las siguientes opciones:

- Use `IngestionMode.BULK` para exportar solo los datos que haya importado de forma masiva mediante un trabajo de importación de conjuntos de datos.
- `IngestionMode.PUT`Utilícelo para exportar únicamente los datos que haya importado de forma individual mediante la consola o las operaciones `PutEvents` PutUsers, o. `PutItems`
- Utilice `IngestionMode.ALL` para exportar todos los datos del conjunto de datos.

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

Requisitos para eliminar recursos de Amazon Personalize

La eliminación de recursos puede ser de ayuda para evitar costos innecesarios. Por ejemplo, se generan costos de campaña mientras una campaña está activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener una lista completa de los cargos y precios, consulte Precios de Amazon Personalize.

Para eliminar recursos con la consola Amazon Personalize, elija Eliminar en la página de detalles del recurso. Para eliminar un recurso con las API de Amazon Personalize, utilice las API `Delete` con los SDK o la AWS Command Line Interface (AWS CLI).

Para ver los pasos detallados para eliminar un conjunto de datos con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS, consulte Eliminación de un conjunto de datos para eliminar todos los datos. Puede aplicar los patrones de estos pasos a otros recursos de Amazon Personalize. Para obtener información sobre cómo eliminar usuarios y los datos del grupo de conjuntos de datos, consulte Eliminación de usuarios y los datos con un trabajo de eliminación de datos.

Debe eliminar algunos recursos para poder eliminar otros. Por ejemplo, si crea un rastreador de eventos y un conjunto de datos de interacciones de elemento, debe eliminar el rastreador de eventos para poder eliminar el conjunto de datos. En las siguientes secciones se indican las directrices y los requisitos de orden para eliminar recursos de Amazon Personalize.

Temas

- Directrices para eliminar recursos
- Orden recomendado para la eliminación de recursos
- Eliminación de usuarios y los datos con un trabajo de eliminación de datos
- Eliminación de un conjunto de datos para eliminar todos los datos

Directrices para eliminar recursos

A continuación, se ofrecen las directrices para eliminar recursos:

- La eliminación de un recurso de Amazon Personalize es una acción irreversible. No se puede detener la eliminación después de que comience.
- No puede eliminar un recurso cuyo estado cambie de un estado a otro. Por ejemplo, no puede eliminar un recurso con el estado CREATE PENDING o IN PROGRESS. El estado del recurso

debe ser ACTIVE (Activo) o CREATE FAILED (Error de creación). Esto incluye el estado `latestSolutionUpdate` de una solución. Puede consultar el estado de un recurso mediante las API `Describe`. Por ejemplo, la operación de la API DescribeCampaign.

- Para obtener información sobre la eliminación de datos de entrenamiento en Amazon S3, consulte ¿Cómo se eliminan objetos de un bucket de S3?
- No se le cobrarán los trabajos de importación de conjuntos de datos una vez finalizados y no podrá eliminarlos.
- No se le cobrarán los esquemas y no podrá eliminar un esquema con la consola de Amazon Personalize. Para eliminar un esquema, utilice la operación de la API DeleteSchema.

A continuación, se indican los requisitos específicos para eliminar conjuntos de datos:

- Debe eliminar todos los filtros antes de eliminar un conjunto de datos.
- Si ha creado un rastreador de eventos, debe eliminarlo antes de eliminar un conjunto de datos de interacciones de elemento.
- Si ha creado una atribución de métrica que hace referencia al conjunto de datos, primero debe eliminar la atribución métrica.
- Si utiliza las recetas User-Personalization-v2, User-Personalization o Next-Best-Action, o bien los casos de uso Mejores opciones y Recomendado para usted, al eliminar un conjunto de datos se detienen las actualizaciones automáticas de cualquier versión de solución o recomendador asociado.
- Ningún `DatasetImportJob` asociado puede tener el estado CREATE PENDING o IN PROGRESS.
- Ningún `BatchInferenceJob` o `BatchSegmentJob` asociado puede tener el estado CREATE PENDING o IN PROGRESS.
- Ningún `Recommender` asociado, `SolutionVersion` puede tener el estado CREATE PENDING o IN PROGRESS.
- Ninguna `Campaign` asociada puede tener el estado CREATE PENDING, IN PROGRESS o ACTIVE.

Orden recomendado para la eliminación de recursos

Para evitar errores de eliminación, le recomendamos que elimine los recursos de un grupo de conjuntos de datos en el siguiente orden. Para identificar los recursos de un grupo de conjuntos de

datos, puede usar las operaciones de la API List. Por ejemplo, puede usar la operación de la API ListFilters para identificar todos los filtros de un grupo de conjuntos de datos.

1. Campañas o recomendadores: para eliminar una campaña o un recomendador con las API, use las operaciones de la API DeleteCampaign o DeleteRecommender. En el caso de los recomendadores, puede detener un recomendador e iniciarlo más tarde. De esta forma, puede pausar la facturación del recomendador y pagarlo solo cuando lo utilice. Para obtener más información, consulte Detención de un recomendador.
2. Soluciones: para eliminar la solución con las API, use la operación de la API DeleteSolution. Para eliminar una solución, no puede estar en curso una actualización de solución. El estado de `latestSolutionUpdate` debe ser ACTIVE o CREATE FAILED. Al eliminar una solución, se eliminan todas las versiones de solución asociadas. Ninguna de las versiones de solución puede tener el estado CREATE PENDING ni IN PROGRESS.
3. Rastreador de eventos: para eliminar un rastreador de eventos con las API, utilice la operación de la API DeleteEventTracker. Debe eliminar el rastreador de eventos, para poder eliminar un conjunto de datos de interacciones de elemento.
4. Atribución de métricas: para eliminar una atribución de métricas con las API, utilice la operación de la API DeleteMetricAttribution.
5. Todos los filtros: para eliminar un filtro con las API, utilice la operación de la API DeleteFilter. Debe eliminar todos los filtros antes de eliminar un conjunto de datos.
6. Conjuntos de datos: para eliminar un conjunto de datos con las API, utilice la operación de la API DeleteDataset.
7. Grupo de conjuntos de datos: para eliminar el grupo de conjuntos de datos con las API, use la operación de la API DeleteDatasetGroup.
8. Esquemas: para eliminar un esquema, use la operación de la API DeleteSchema.

Eliminación de usuarios y los datos con un trabajo de eliminación de datos

Después de importar los datos, puede eliminar los usuarios y los datos, incluidos los metadatos y los datos de interacciones, de un grupo de conjuntos de datos. Puede eliminar los datos de usuario como parte de un programa de cumplimiento, para atender las solicitudes de eliminación de usuarios o para mantener los datos actualizados a medida que cambie la base de usuarios.

Después de eliminar los usuarios, Amazon Personalize ya no utiliza los datos para el entrenamiento y deja de tener en cuenta a los usuarios al generar segmentos de usuarios.

Para eliminar las referencias a los usuarios en los conjuntos de datos y modelos de Amazon Personalize de un grupo de conjuntos de datos, haga lo siguiente:

1. Prepare un archivo CSV que enumere los ID de usuario de los usuarios que desee eliminar en una columna USER_ID.
2. Cargue el archivo CSV en un bucket de Amazon S3. El rol de servicio de Amazon Personalize debe tener permiso para acceder a este bucket.
3. Cree un trabajo de eliminación de datos. Un trabajo de eliminación de datos es un trabajo por lotes que elimina los usuarios y los datos de los modelos y conjuntos de datos de un grupo de conjuntos de datos.

Temas

- Directrices y requisitos
- Preparación de una lista de usuarios para eliminarlos
- Creación de un trabajo de eliminación de datos

Directrices y requisitos

A continuación, se muestran las directrices y los requisitos para eliminar usuarios:

- Antes de crear un trabajo de eliminación de datos, asegúrese de que no haya ningún trabajo en curso que use los conjuntos de datos, como trabajos de entrenamiento, trabajos por lotes u operaciones de importación masiva o individual. Y evite crear dichos trabajos mientras haya un trabajo de eliminación de datos en curso. Si se produce un entrenamiento o una importación, no podemos garantizar que los datos de los usuarios se eliminen de los modelos, por lo que recomendamos crear un trabajo de eliminación de datos adicional.
- Un trabajo de eliminación de datos no elimina las referencias a usuarios que no estén en Amazon Personalize. Por ejemplo, no elimina el ID de usuario de las recomendaciones por lotes del bucket de Amazon S3. Debe eliminar estos registros manualmente.
- Puede tener hasta cinco trabajos de eliminación para un grupo de conjuntos de datos con el estado PENDING.

- El tamaño total máximo de los archivos de entrada de eliminación de datos es de 100 MB. Puede reutilizar el mismo archivo de entrada cuando crea trabajos de eliminación.
- Cada trabajo de eliminación de datos elimina los usuarios y los datos de interacciones en un grupo de conjuntos de datos. Para eliminar los datos en todos los grupos de conjuntos de datos, debe crear un trabajo de eliminación de datos para cada grupo de conjuntos de datos.
- Después de crear un trabajo, es posible que se tarde hasta un día en eliminar los datos de los usuarios de los conjuntos de datos y modelos.
- Después de que se complete un trabajo, asegúrese de actualizar los recursos personalizados. Asegúrese de crear una nueva versión de solución y, si es necesario, actualice la campaña. Si utiliza el entrenamiento automático, aún puede crear nuevas versiones de solución de forma manual.
- El rol de servicio de Amazon Personalize debe tener permiso para acceder al bucket de Amazon S3 con la lista de los usuarios que se eliminarán. Necesita los permisos `GetObject` y `ListBucket` para el bucket y el contenido. Estos permisos son los mismos que en la importación de datos. Para obtener información sobre la concesión de permisos y ejemplos de políticas, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.
- No puede usar la clave AWS Key Management Service en el bucket de Amazon S3 que almacena la lista de ID de los usuarios que desea eliminar.
- Si un elemento aparece únicamente en el conjunto de datos de interacciones de elemento y solo los usuarios que va a eliminar han interactuado con él, este elemento dejará de aparecer en las recomendaciones.

Preparación de una lista de usuarios para eliminarlos

Antes de eliminar usuarios de Amazon Personalize, debe preparar una lista de los usuarios que se eliminarán en un archivo CSV y cargarla en Amazon S3.

Preparación y carga de la lista de los usuarios que se eliminarán

1. Cree un archivo CSV que enumere los ID de los usuarios que se eliminarán. A continuación, se muestra cómo se debe formatear el archivo CSV.

```
USER_ID
abc
2a
5basc
ab35
```

```
123f
a55d
0v22
441fa
efg
```

2. Cargue el archivo CSV en un bucket de Amazon Simple Storage Service (Amazon S3). Para obtener más información sobre la carga de archivos en Amazon S3, consulte Carga de archivos y carpetas con la función arrastrar y soltar en la Guía del usuario de Amazon Simple Storage Service.
3. Conceda acceso a Amazon Personalize al bucket y el archivo CSV. Amazon Personalize debe tener permiso para realizar las acciones `GetObject` y `ListBucket` en el bucket y el contenido. Estos permisos son los mismos que en la importación de datos. Para obtener información sobre la concesión de permisos y ejemplos de políticas, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Creación de un trabajo de eliminación de datos

Después de completar Preparación de una lista de usuarios para eliminarlos, podrá eliminar los usuarios con un trabajo de eliminación de datos.

Un trabajo de eliminación de datos es un trabajo por lotes que elimina los usuarios y los datos de los modelos y conjuntos de datos de un grupo de conjuntos de datos. Después de eliminar los usuarios, Amazon Personalize ya no utiliza los datos para el entrenamiento y deja de tener en cuenta a los usuarios al generar segmentos de usuarios.

Cuando cree un trabajo de eliminación de datos, especifique la ubicación de Amazon S3 de la lista de los usuarios que se eliminarán.

- Si los datos se encuentran en un solo archivo, utilice la siguiente sintaxis para la ubicación de Amazon S3:

  `s3://amzn-s3-demo-bucket/<folder path>/<CSV filename>.csv`
- Si los archivos CSV se encuentran en una carpeta del bucket de Amazon S3, puede especificar la ruta a la carpeta. Con una tarea de eliminación de datos, Amazon Personalize utiliza todos los archivos con la extensión de archivo `.csv` de la carpeta y de cualquier subcarpeta. Ignora los archivos de otro tipo. Use la siguiente sintaxis con una `/` después del nombre de la carpeta:

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

El rol que utilice debe tener permiso para realizar las acciones `GetObject` y `ListBucket` en el bucket de Amazon S3 y el contenido. Para obtener información sobre la concesión de permisos y ejemplos de políticas, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Puede crear un trabajo de eliminación de datos con la consola de Amazon Personalize, la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Creación de un trabajo de eliminación de datos (consola)

Para eliminar usuarios con la consola de Amazon Personalize, cree un trabajo de eliminación de datos con un nombre, el rol de servicio de IAM y la ubicación de los datos en Amazon S3.

Eliminación de registros (consola)

1. Abra la consola de Amazon Personalize en https://console.aws.amazon.com/personalize/home e inicie sesión en su cuenta.
2. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos. Aparece la Información general del grupo de conjuntos de datos.
3. En el panel de navegación de la izquierda, elija Conjuntos de datos.
4. En Trabajos de eliminación de datos, elija Crear trabajo.
5. En Detalles del trabajo, asigne un nombre al trabajo.
6. En Origen de entrada de S3, para Ubicación de S3, especifique la ubicación en Amazon S3 del archivo CSV que almacena la lista de los ID de los usuarios que se van a eliminar. Este archivo lo ha preparado en Preparación de una lista de usuarios para eliminarlos.
7. En Rol de IAM, elija crear un rol nuevo o usar uno existente. Si ha completado los requisitos previos para crear un rol para Amazon Personalize y ha concedido acceso a este rol al bucket de Amazon S3, elija Usar un rol de servicio existente y especifique el rol que ha creado en Creación de un rol de IAM para Amazon Personalize.

   El rol que utilice debe tener permiso para realizar las acciones `GetObject` y `ListBucket` en el bucket de Amazon S3 y el contenido. Estos permisos son los mismos que en la importación de datos. Para obtener información sobre la concesión de permisos y ejemplos de políticas, consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.
8. Para Etiquetas, si lo desea, añada cualquier etiqueta. Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.

9. Seleccione Crear tarea. El trabajo comienza y se muestra la página de detalles.

   Después de crear un trabajo, puede tardar hasta un día en eliminar los datos de los usuarios de los conjuntos de datos y modelos. Hasta que se complete el trabajo, Amazon Personalize seguirá utilizando los datos durante el entrenamiento. Además, los usuarios pueden aparecer en segmentos de usuarios.

   La eliminación de datos finaliza cuando el estado es COMPLETED. Si se produce un error en el trabajo por algún motivo, recomendamos crear otro trabajo de eliminación de datos. Después de que se complete un trabajo, asegúrese de actualizar los recursos personalizados. Asegúrese de crear una nueva versión de solución y, si es necesario, actualice la campaña. Si utiliza el entrenamiento automático, aún puede crear nuevas versiones de solución de forma manual.

Creación de un trabajo de eliminación de datos (AWS CLI)

Para eliminar usuarios con la AWS CLI, ejecute el comando `create-data-deletion-job`. Este comando usa la operación de la API CreateDataDeletion. En el siguiente código se muestra cómo crear un trabajo de eliminación de datos. Para usar el código, actualícelo para especificar el nombre del trabajo, el rol de IAM que ha creado en Creación de un rol de IAM para Amazon Personalize y la ubicación de los datos en Amazon S3. Este archivo lo ha preparado en Preparación de una lista de usuarios para eliminarlos.

```
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
```

Después de crear un trabajo, puede tardar hasta un día en eliminar los datos de los usuarios de los conjuntos de datos y modelos. Hasta que se complete el trabajo, Amazon Personalize seguirá utilizando los datos durante el entrenamiento. Además, los usuarios pueden aparecer en segmentos de usuarios.

El trabajo finaliza cuando el estado es COMPLETED. Consulte el estado mediante el comando `describe-data-deletion-job` y especifique el ARN del trabajo de eliminación de datos. Para obtener más información sobre la operación de la API, consulte DescribeDataDeletionJob. Para ver un historial de los trabajos de eliminación de datos ordenados por hora de creación, utilice la operación de la API ListDataDeletionJobs.

Si se produce un error en el trabajo por algún motivo, recomendamos crear otro trabajo de eliminación de datos. Después de que se complete un trabajo, asegúrese de actualizar los recursos personalizados. Asegúrese de crear una nueva versión de solución y, si es necesario, actualice la campaña. Si utiliza el entrenamiento automático, aún puede crear nuevas versiones de solución de forma manual.

Creación de un trabajo de eliminación de datos (SDK de AWS)

Para eliminar usuarios con los SDK de AWS, utilice la operación de la API CreateDataDeletionJob. En el siguiente código se muestra cómo crear un trabajo de eliminación de datos. Para usar el código, actualícelo para especificar el nombre del trabajo, el rol de IAM que ha creado en Creación de un rol de IAM para Amazon Personalize y la ubicación de los datos en Amazon S3. Este archivo lo ha preparado en Preparación de una lista de usuarios para eliminarlos.

```
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'])
```

Después de crear un trabajo, puede tardar hasta un día en eliminar los datos de los usuarios de los conjuntos de datos y modelos. Hasta que se complete el trabajo, Amazon Personalize seguirá utilizando los datos durante el entrenamiento. Además, los usuarios pueden aparecer en segmentos de usuarios.

El trabajo finaliza cuando el estado es COMPLETED. Consulte el estado mediante la operación DescribeDataDeletionJob y especifique el ARN del trabajo de eliminación de datos. Para ver un historial de los trabajos de eliminación de datos ordenados por hora de creación, utilice la operación de la API ListDataDeletionJobs.

Si se produce un error en el trabajo por algún motivo, recomendamos crear otro trabajo de eliminación de datos. Después de que se complete un trabajo, asegúrese de actualizar los recursos personalizados. Asegúrese de crear una nueva versión de solución y, si es necesario, actualice la campaña. Si utiliza el entrenamiento automático, aún puede crear nuevas versiones de solución de forma manual.

Eliminación de un conjunto de datos para eliminar todos los datos

Para eliminar todos los datos en un conjunto de datos, elimine el conjunto de datos. Puede eliminar un conjunto de datos con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS SDKs. Antes de eliminar un conjunto de datos, tenga en cuenta lo siguiente:

- Debe eliminar todos los filtros antes de eliminar un conjunto de datos.
- Si ha creado un rastreador de eventos, debe eliminarlo antes de eliminar un conjunto de datos de interacciones de elemento.
- Si ha creado una atribución de métrica que hace referencia al conjunto de datos, primero debe eliminar la atribución métrica.
- Si utilizas User-Personalization-v dos casos de uso: personalización de usuarios, Next-Best-Action recetas o las mejores opciones para ti y las recomendadas para ti, al eliminar un conjunto de datos se detienen las actualizaciones automáticas de cualquier versión o recomendación de la solución asociada.
- Ningún `DatasetImportJob` asociado puede tener el estado CREATE PENDING o IN PROGRESS.
- Ningún `BatchInferenceJob` o `BatchSegmentJob` asociado puede tener el estado CREATE PENDING o IN PROGRESS.
- Ningún `Recommender` asociado, `SolutionVersion` puede tener el estado CREATE PENDING o IN PROGRESS.
- Ninguna `Campaign` asociada puede tener el estado CREATE PENDING, IN PROGRESS o ACTIVE.

Temas

- Eliminación de un conjunto de datos (consola)
- Eliminación de un conjunto de datos (AWS CLI)
- Eliminación de un conjunto de datos (AWS SDKs)

Eliminación de un conjunto de datos (consola)

Para eliminar un conjunto de datos con la consola de Amazon Personalize, vaya a la página de detalles del conjunto de datos y elija eliminar.

Para eliminar un conjunto de datos

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa.
2. En el panel de navegación, elija Grupos de conjuntos de datos.
3. En la página Grupos de conjuntos de datos, elija su grupo de conjuntos de datos.
4. Elija Conjuntos de datos en el panel de navegación.
5. Elija el conjunto de datos para abrir la página de detalles.
6. En la página de detalles del conjunto de datos, elija Eliminar y confirme la eliminación del conjunto de datos.

Eliminación de un conjunto de datos (AWS CLI)

El siguiente código muestra cómo eliminar un conjunto de datos con la DeleteDataset operación AWS CLI y.

```
aws personalize delete-dataset --dataset-arn dataset-arn
```

Eliminación de un conjunto de datos (AWS SDKs)

El siguiente código muestra cómo eliminar un conjunto de datos con la DeleteDataset operación AWS SDKs y.

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());
    }
}
```

Medición del impacto de las recomendaciones de Amazon Personalize

A medida que los clientes interactúan con las recomendaciones, puede medir cómo lo ayudan las recomendaciones de Amazon Personalize a alcanzar los objetivos. Puede identificar qué campañas y recomendadores tienen el mayor impacto en las métricas de rendimiento clave. Por ejemplo, puede identificar qué recurso genera el máximo de minutos de visualización, clics o compras. Además, puedes comparar el rendimiento de las recomendaciones de Amazon Personalize con las generadas por servicios de terceros.

Cuando sepa qué campaña o qué recomendador genera el mayor impacto, podrá tomar medidas para aprovechar aún más las recomendaciones. Por ejemplo, puede aumentar la visibilidad de las recomendaciones en el sitio para generar más interacciones. O puede incluir las recomendaciones en una campaña de marketing, como correos electrónicos personalizados o anuncios segmentados.

Si identifica un recurso que no tiene el impacto esperado, puede tomar medidas para mejorar las recomendaciones. Por ejemplo, puede usar la consola de Amazon Personalize para analizar los datos de entrenamiento utilizados para crear el recurso, realizar las mejoras de datos recomendadas y, a continuación, volver a importar los datos. Para obtener más información sobre el análisis de datos, consulte Análisis de la calidad y la cantidad de datos en los conjuntos de datos de Amazon Personalize.

Lo siguiente puede ayudarte a medir el impacto de las recomendaciones:

- Atribución de métricas: una atribución de métricas de Amazon Personalize crea informes basados en las métricas que especifique y en los datos e interacciones de elementos que importe. Por ejemplo, la duración total de las películas vistas por los usuarios o el número total de eventos de clic. Tras crear una atribución de métricas, Amazon Personalize envía automáticamente las métricas sobre los eventos de la operación de la PutEvents API y los datos incrementales masivos a Amazon CloudWatch. Para los datos masivos, puede optar por publicar informes en un bucket de Amazon S3.
- Pruebas A/B: realizar A/B pruebas con las recomendaciones de Amazon Personalize implica mostrar diferentes tipos de recomendaciones a diferentes grupos de usuarios y comparar los resultados. Puede utilizar A/B las pruebas para comparar y evaluar diferentes estrategias de recomendación, evaluar el rendimiento del modelo y medir el impacto de las recomendaciones.

Temas

- Medición del impacto de las recomendaciones con una atribución de métricas
- Medir el impacto de las recomendaciones mediante A/B pruebas

Medición del impacto de las recomendaciones con una atribución de métricas

Para medir el impacto de las recomendaciones de elementos, puede crear una atribución de métricas. Una atribución de métricas crea informes basados en los datos e interacciones de elementos que importe, así como en las métricas que especifique. Por ejemplo, la duración total de las películas vistas por los usuarios o el número total de eventos de clic. Amazon Personalize agrega los cálculos en un período de 15 minutos. Para los datos de interacciones transmitidos y los datos masivos incrementales, Amazon Personalize envía automáticamente informes de métricas a Amazon CloudWatch. Para los datos masivos, puede optar por publicar informes en un bucket de Amazon S3.

Para cada interacción que importe, incluya datos de origen para comparar diferentes campañas, generadores de recomendaciones y terceros. Puede incluir el ID de recomendación de las recomendaciones que mostró al usuario o al origen de eventos, por ejemplo, a un tercero.

Por ejemplo, es posible que tenga una aplicación de streaming de vídeo que muestre recomendaciones de películas de dos generadores de recomendaciones diferentes de Amazon Personalize. Si desea ver qué generador de recomendaciones genera el mayor número de eventos de visualización, puede crear una atribución de métricas que realice un seguimiento del número total de eventos de visualización. A continuación, podría registrar los eventos de visualización a medida que los usuarios interactúen con las recomendaciones e incluir `recommendationId` en cada evento. Amazon Personalize utiliza el `recommendationId` para identificar a cada generador de recomendaciones. A medida que grabas los eventos, puedes ver los totales de los eventos vistos acumulados cada 15 minutos para los dos usuarios que realizaron la recomendación. CloudWatch Para ver ejemplos de código que muestran cómo incluir un `recommendationId` o un `eventAttributionSource` para un evento, consulte Informes de atribución y métricas de eventos.

Temas

- Directrices y requisitos para una atribución de métricas
- Creación de una atribución de métricas de Amazon Personalize
- Actualización de una atribución de métricas de Amazon Personalize
- Eliminación de una atribución de métricas de Amazon Personalize

- Visualización de gráficos de datos métricos en CloudWatch
- Publicación de informes de atribución de métricas en Amazon S3

Directrices y requisitos para una atribución de métricas

Amazon Personalize comienza a calcular e informar del impacto de las recomendaciones solo después de crear una atribución de métricas. Para crear el historial más completo, le recomendamos que cree una atribución de métricas antes de importar sus datos de interacciones. Al crear un trabajo de importación de conjuntos de datos para un conjunto de datos de interacciones de elementos con la consola de Amazon Personalize, tiene la opción de crear una atribución de métricas en una pestaña nueva. A continuación, puede volver al trabajo de importación para completarlo.

Tras crear una atribución métrica y registrar los eventos o importar datos masivos incrementales, incurrirás en un coste mensual CloudWatch por métrica. Para obtener información sobre CloudWatch los precios, consulta la página de CloudWatch precios de Amazon. Para dejar de enviar métricas a CloudWatch, elimina la atribución de métricas.

Para ver el impacto de las recomendaciones a lo largo del tiempo, siga importando datos a medida que los clientes interactúen con recomendaciones. Si ya ha importado datos, puede seguir creando una atribución de métricas y empezar a medir el impacto de las recomendaciones. Sin embargo, Amazon Personalize no informará sobre los datos que haya importado antes de crearlos.

A continuación se indican las directrices y los requisitos para generar informes con una atribución de métricas:

- Debes conceder permiso a Amazon Personalize para acceder a los datos y colocarlos CloudWatch. Para ver ejemplos de políticas, consulte Otorgar a Amazon Personalize acceso a CloudWatch.
- Para publicar métricas en Amazon S3, conceda permiso a Amazon Personalize para escribir en su bucket. También debe proporcionar la ruta del bucket en la atribución de métricas. Para ver ejemplos de políticas, consulte Concesión de acceso a Amazon Personalize para su bucket de Amazon S3.
- Para publicar las métricas en CloudWatch, los registros deben tener menos de 14 días de antigüedad. Si sus datos son más antiguos, estos registros no se incluirán en los cálculos ni en los informes.
- La importación de eventos duplicados (eventos que coinciden exactamente con todos los atributos) puede provocar un comportamiento inesperado, como métricas imprecisas. Recomendamos

eliminar los registros duplicados de los datos masivos antes de importarlos, y evitar importar eventos duplicados con la operación `PutEvents`.

- El conjunto de datos de interacciones de elementos debe tener una columna `EVENT_TYPE`.
- No puede crear informes de métricas para los datos de un conjunto de datos de interacciones de acción.
- Puede crear una atribución de métricas como máximo por grupo de conjuntos de datos. Cada atribución de métricas puede tener 10 métricas como máximo.

Para comparar los orígenes, cada evento de interacción debe incluir una `recommendationId` o`eventAttributionSource`. Puede importar como máximo 100 orígenes de atribución de eventos únicos. Para obtener ejemplos de código de `PutEvents`, consulte Informes de atribución y métricas de eventos.

- Si proporciona `recommendationId`, Amazon Personalize determina automáticamente la campaña o el recomendador de origen y lo identificará en los informes de una columna EVENT_ATTRIBUTION_SOURCE.
- Si proporciona ambos atributos, Amazon Personalize utilizará únicamente `eventAttributionSource`.
- Si no proporciona un origen, Amazon Personalize etiqueta el origen `SOURCE_NAME_UNDEFINED` en los informes.

Temas

- Otorgar a Amazon Personalize acceso a CloudWatch
- Concesión de acceso a Amazon Personalize para su bucket de Amazon S3

Otorgar a Amazon Personalize acceso a CloudWatch

> **Important**
>
> Cuando concedes permisos, Amazon Personalize coloca y valida una pequeña cantidad de datos. CloudWatch Esto conllevará un coste único de menos de 0,30 $. Para obtener más información sobre CloudWatch los precios, consulta la página de CloudWatch precios de Amazon.

Para dar acceso a Amazon Personalize CloudWatch, adjunta una nueva política AWS Identity and Access Management (IAM) a tu rol de servicio Amazon Personalize que otorgue al rol permiso para CloudWatch usar la `PutMetricData` Acción. La siguiente política de ejemplo concede permisos de `PutMetricData`.

JSON

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

Concesión de acceso a Amazon Personalize para su bucket de Amazon S3

Para conceder acceso a Amazon Personalize para su bucket de Amazon S3:

- Asocie una política de IAM a su rol de servicio de Amazon Personalize que conceda al rol permiso para usar la acción `PutObject` en su bucket.

  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/*"
            ]
        }
    ]
}
```

- Asocie una política de buckets a su bucket de salida de Amazon S3 que conceda permiso de principio de Amazon Personalize para usar las acciones de `PutObject`.

  Si utilizas AWS Key Management Service (AWS KMS) para el cifrado, debes conceder permiso a Amazon Personalize y a tu rol de servicio Amazon Personalize IAM para usar tu clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

  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/*"
            ]
        }
    ]
}
```

Creación de una atribución de métricas de Amazon Personalize

> **Important**
>
> Tras crear una atribución de métrica y registrar los eventos o importar datos masivos incrementales, incurrirá en un CloudWatch coste mensual por métrica. Para obtener información sobre CloudWatch los precios, consulta la página de CloudWatch precios de Amazon. Para dejar de enviar métricas a CloudWatch, elimina la atribución de métricas.

Para empezar a generar informes de métricas, debe crear una atribución de métricas e importar los datos de interacciones. Al crear una atribución de métricas, se especifica una lista de tipos de eventos sobre los que informar. Para cada tipo de evento, debe especificar una función que Amazon Personalize aplique a medida que recopila los datos. Entre las funciones disponibles se incluyen `SUM(DatasetType.COLUMN_NAME)` y `SAMPLECOUNT()`.

Por ejemplo, es posible que tenga una aplicación de streaming de vídeo en línea y desee realizar un seguimiento de dos métricas: la tasa de clics para recomendaciones y la duración total de las películas vistas, donde cada vídeo del conjunto de datos de elementos incluye un atributo `LENGTH`. Debería crear una atribución de métricas y añadir dos métricas, cada una de ellas con un tipo de evento y una función. La primera podría ser para el tipo de evento `Click` con una función `SAMPLECOUNT()`. La segunda podría ser para el tipo de evento `Watch` con una función `SUM(Items.LENGTH)`.

Puede aplicar funciones `SUM()` solo a las columnas numéricas de los conjuntos de datos de elementos y de interacciones de elementos. Para aplicar una función `SUM()` a una columna de un conjunto de datos de elementos, primero debe importar los metadatos de los elementos.

Puede crear una atribución de métricas con la consola Amazon Personalize o AWS Command Line Interface los AWS SDK.

Temas

- Creación de una atribución de métricas (consola)
- Crear una atribución métrica (AWS CLI)
- Crear una atribución métrica (AWS SDK)

Creación de una atribución de métricas (consola)

Para crear una atribución de métricas con la consola de Amazon Personalize, vaya a la página Atribución de métricas y elija Crear atribución de métricas. Cuando cree una atribución de métricas, debe especificar una ruta de bucket de Amazon S3 opcional, su rol de servicio de IAM de Amazon Personalize y una lista de métricas sobre las que informar.

Al crear un trabajo de importación de conjuntos de datos de interacciones de elementos con la consola de Amazon Personalize, tiene la opción de crear una atribución de métricas en una pestaña nueva. A continuación, puede volver al trabajo de importación para completarlo. Si ya se encuentra en la página Configurar atribución de métricas, puede ir directamente al paso 4.

Para crear una atribución de métricas

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija su grupo de conjuntos de datos.
3. En el panel de navegación, en Recursos personalizados, elija Atribución de métricas.
4. En Detalles de atribución de métricas, elija Crear atribución de métricas.
5. En la página Configurar atribución de métricas, asigne un nombre a la atribución de métricas.
6. Si desea publicar métricas en Amazon S3 para la Ruta de salida de datos de Amazon S3, especifique el bucket de Amazon S3 de destino. Esto habilita la opción de publicar métricas cada vez que cree un trabajo de importación de conjuntos de datos. Utilice la siguiente sintaxis:

   `s3://amzn-s3-demo-bucket/<folder> path>`
7. Si lo utiliza AWS KMS para el cifrado, para el ARN de la clave de KMS, introduzca el nombre de recurso de Amazon (ARN) de la clave. AWS KMS Debe conceder permiso a Amazon Personalize y a su rol de servicio de IAM de Amazon Personalize para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.
8. En Rol de IAM, elija entre crear un rol de servicio nuevo o usar uno que ya exista. El rol que elija debe tener `PutMetricData` permisos. CloudWatch Si desea publicar en Amazon S3, el rol debe tener permisos de `PutObject` para su bucket de Amazon S3.

   Para usar el rol en el que creóCreación de un rol de IAM para Amazon Personalize, es posible que tenga que añadir políticas para CloudWatch Amazon S3.

Para ver más ejemplos de política, consulte Otorgar a Amazon Personalize acceso a CloudWatch y Concesión de acceso a Amazon Personalize para su bucket de Amazon S3.

9. Elija Siguiente.
10. En la página Definir atributos de métricas, elija cómo definir las métricas. Elija Crear atributos de métricas para usar la herramienta del compilador. Elija Introducir atributos de métrica para introducir las métricas en formato JSON.

    - Si eliges Crear atributos de métrica, proporcione un nombre, un tipo de evento y elija una función para cada métrica. Para las funciones `SUM()`, elija el nombre de la columna. Seleccione Añadir atributo de métrica para añadir métricas adicionales.
    - Si elige Introducir atributos de métricas, introduzca cada métrica en formato JSON. A continuación se muestra cómo dar formato a una métrica.

    ```
    {
        "EventType": "watch",
        "MetricName": "MinutesWatchedTracker",
        "MetricMathExpression": "SUM(Items.LENGTH)"
    }
    ```

11. Elija Siguiente.
12. En la página Revisar y crear página, revise los detalles de la nueva atribución de métricas. Para realizar cambios, seleccione Previous (Anterior). Para crear la atribución de métricas, elija Crear. Cuando la atribución de métricas esté activa, puede empezar a importar datos y ver los resultados. Para obtener más información acerca de la visualización de resultados, consulte Visualización de gráficos de datos métricos en CloudWatch. Para obtener información sobre la publicación de resultados en Amazon S3, consulte Publicación de informes de atribución de métricas en Amazon S3.

Crear una atribución métrica (AWS CLI)

El código siguiente muestra cómo crear una atribución de métricas con AWS Command Line Interface. El rol que especifique debe tener `PutMetricData` permisos CloudWatch y, si publica en Amazon S3, `PutObject` permisos para su bucket de Amazon S3. Para usar el rol en el que creóCreación de un rol de IAM para Amazon Personalize, es posible que tenga que añadir políticas para CloudWatch Amazon S3. Para ver más ejemplos de política, consulte Otorgar a Amazon

Personalize acceso a CloudWatch y Concesión de acceso a Amazon Personalize para su bucket de Amazon S3.

Para cada métrica, especifique un nombre, un tipo de evento y una expresión (una función). Entre las funciones disponibles se incluyen `SUM(DatasetType.COLUMN_NAME)` y `SAMPLECOUNT()`. Para las funciones SUM(), especifique el tipo de conjunto de datos y el nombre de la columna. Por ejemplo, `SUM(Items.LENGTH)`. Para obtener información acerca de cada parámetro, consulte CreateMetricAttribution.

```
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\"
}]"
```

Crear una atribución métrica (AWS SDK)

El código siguiente muestra cómo crear una atribución de métricas con AWS SDK para Python (Boto3). El rol que especifique debe tener `PutMetricData` permisos CloudWatch y, si publica en Amazon S3, `PutObject` permisos para su bucket de Amazon S3. Para usar el rol en el que creóCreación de un rol de IAM para Amazon Personalize, es posible que tenga que añadir políticas para CloudWatch Amazon S3. Para ver más ejemplos de política, consulte Otorgar a Amazon Personalize acceso a CloudWatch y Concesión de acceso a Amazon Personalize para su bucket de Amazon S3.

Para cada métrica, especifique un nombre, un tipo de evento y una expresión (una función). Entre las funciones disponibles se incluyen `SUM(DatasetType.COLUMN_NAME)` y `SAMPLECOUNT()`. Para las funciones SUM(), especifique el tipo de conjunto de datos y el nombre de la columna. Por ejemplo, `SUM(Items.LENGTH)`. Para obtener información acerca de cada parámetro, consulte CreateMetricAttribution.

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();
```

Actualización de una atribución de métricas de Amazon Personalize

Al actualizar una atribución de métricas, puede añadir y eliminar métricas, así como modificar su configuración de salida. Puede actualizar la atribución de una métrica con la consola Amazon Personalize o AWS Command Line Interface los AWS SDK.

Temas

- Actualización de una atribución de métricas (consola)
- Actualizar la atribución de una métrica (AWS CLI)
- Actualizar una atribución de métrica (AWS SDK)

Actualización de una atribución de métricas (consola)

Para actualizar una atribución de métricas con la consola de Amazon Personalize, realice los cambios en la página Atribución de métricas.

Para actualizar una atribución de métricas

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija su grupo de conjuntos de datos.
3. En el panel de navegación, elija Métricas.
4. En la sección inferior, elija la pestaña Atributos de métricas o la pestaña Configuración de atribuciones de métricas para empezar a realizar cambios.

    - Para añadir o eliminar métricas, seleccione la pestaña Atributos de métricas y elija Editar atributos. Realice los cambios en la página Editar atributos de métricas y elija Actualizar para guardar los cambios.
    - Para realizar cambios en el bucket de salida de Amazon S3 o en el rol de servicio de IAM, seleccione la pestaña Editar configuración de atribución de métricas y realice los cambios en la página Editar configuración de atribuciones. Elija Actualizar para guardar los cambios.

Actualizar la atribución de una métrica (AWS CLI)

Tras crear una atribución de métrica, puede utilizar AWS Command Line Interface (AWS CLI) para añadir y eliminar métricas y modificar su configuración de salida. El código siguiente muestra cómo eliminar métricas con el comando `update-metric-attribution`:

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--remove-metrics metricName1 metricName2
```

El siguiente código muestra cómo añadir una métrica adicional y especificar una nueva configuración de salida:

```
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\"
}]"
```

Si se ejecuta correctamente, Amazon Personalize devuelve el ARN de la atribución de métricas que ha actualizado. Para obtener un listado completo de todos los parámetros, consulte UpdateMetricAttribution.

Actualizar una atribución de métrica (AWS SDK)

Después de crear una atribución de métricas, puede agregar o quitar métricas, así como modificar su configuración de salida. El código siguiente muestra cómo eliminar métricas de una atribución de métricas.

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();
```

El siguiente código muestra cómo añadir una métrica adicional y especificar una nueva configuración de salida:

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();
```

Si se ejecuta correctamente, Amazon Personalize devuelve el ARN de la atribución de métricas que ha actualizado. Para obtener un listado completo de todos los parámetros, consulte UpdateMetricAttribution.

Eliminación de una atribución de métricas de Amazon Personalize

Si ya no desea generar informes, puede eliminar una atribución de métricas. Al eliminar una atribución de métricas, se eliminan todas sus métricas y su configuración de salida.

Si eliminas una atribución de métrica, Amazon Personalize deja de enviar automáticamente los informes relacionados con los datos masivos incrementales PutEvents y a CloudWatch ellos. Los datos ya enviados CloudWatch o publicados en Amazon S3 no se ven afectados. Puedes eliminar una atribución de métrica con la consola Amazon Personalize o AWS Command Line Interface los AWS SDK.

Temas

- Eliminación de una atribución de métricas (consola)
- Eliminar una atribución de métrica (AWS CLI)

- Eliminar una atribución métrica (AWS SDK)

Eliminación de una atribución de métricas (consola)

Elimine una atribución de métricas en la página de información general para su atribución de métricas.

Para eliminar una atribución de métricas

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/homee inicia sesión en tu cuenta.
2. Elija su grupo de conjuntos de datos.
3. En el panel de navegación, elija Métricas.
4. Elija Eliminar y, a continuación, confirme la eliminación.

Eliminar una atribución de métrica (AWS CLI)

Para eliminar una atribución de métrica con el AWS CLI, utilice el `delete-metric-attribution` comando de la siguiente manera.

```
aws personalize delete-metric-attribution --metric-attribution-arn metric attribution
 ARN
```

Eliminar una atribución métrica (AWS SDK)

El código siguiente muestra cómo eliminar una atribución de métricas con el SDK para 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();
```

Visualización de gráficos de datos métricos en CloudWatch

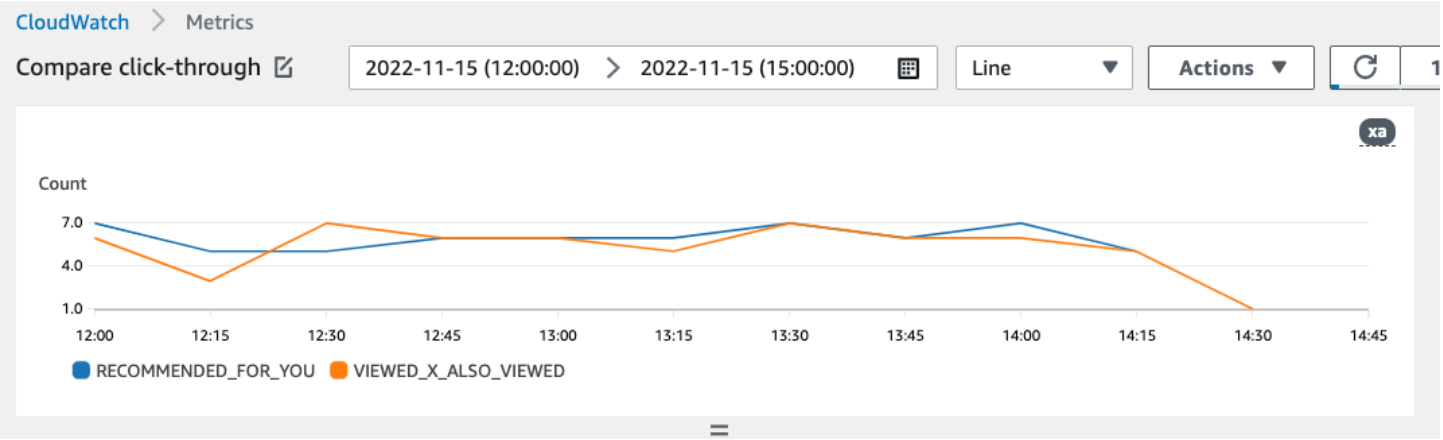
> **Important**
>
> Tras crear una atribución de métricas y registrar los eventos o importar datos incrementales de forma masiva, incurrirás en un CloudWatch coste mensual por métrica. Para obtener información sobre CloudWatch los precios, consulta la página de CloudWatch precios de Amazon. Para dejar de enviar métricas a CloudWatch, elimina la atribución de métricas.

Tras crear una atribución de métricas, Amazon Personalize envía automáticamente las métricas PutEvents y los datos incrementales masivos a Amazon CloudWatch. Puede seleccionar métricas y crear gráficos de los datos de las métricas mediante la CloudWatch consola. Estos gráficos pueden ayudarlo a inspeccionar visualmente y comparar el rendimiento y el impacto de diferentes recomendadores o campañas.

Para comparar los orígenes, cada evento de interacción debe incluir una `recommendationId` o`eventAttributionSource`. Para ver muestras de código que presentan cómo incluir estos en un evento, consulte Informes de atribución y métricas de eventos.

Para ver las métricas CloudWatch, complete el procedimiento que se encuentra en Graficar una métrica. Puede ver los datos con diferentes niveles de detalle. El Período mínimo que puede representar en una gráfica es de 15 minutos. Puede ver los datos de Amazon Personalize de las 2 semanas anteriores en CloudWatch ; los datos más antiguos se ignoran. Para el término de búsqueda, especifique el nombre que le dio a la métrica cuando creó la atribución de métricas.

A continuación se muestra un ejemplo de cómo puede aparecer una métrica en CloudWatch. La métrica muestra la tasa de clics por cada 15 minutos para dos generadores de recomendaciones diferentes.



Publicación de informes de atribución de métricas en Amazon S3

Para todos los datos masivos, si proporciona un bucket de Amazon S3 al crear su atribución de métricas, puede optar por publicar informes de métricas en su bucket de Amazon S3 cada vez que cree un trabajo de importación de conjuntos de datos para los datos de interacciones.

Para publicar métricas en Amazon S3, debe proporcionar una ruta a su bucket de Amazon S3 en la atribución de métricas. A continuación, publique los informes en Amazon S3 cuando cree un trabajo de importación de conjuntos de datos. Cuando se complete el trabajo, podrá encontrar las métricas en su bucket de Amazon S3. Cada vez que publique métricas, Amazon Personalize crea un archivo nuevo en su bucket de Amazon S3. El nombre del archivo incluye el método de importación y la fecha de la siguiente manera:

`AggregatedAttributionMetrics - ImportMethod - Timestamp.csv`

A continuación se muestra un ejemplo de cómo pueden aparecer las primeras filas de un archivo CSV de informe de métricas. La métrica de este ejemplo indica el total de clics de dos generadores de recomendaciones diferentes en intervalos de 15 minutos. Cada generador de recomendaciones se identifica por su nombre de recurso de Amazon (ARN) en la columna EVENT_ATTRIBUTION_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
......
.....
```

Publicación de métricas para datos masivos en Amazon S3 (consola)

Para publicar métricas en un bucket de Amazon S3 con la consola de Amazon Personalize, cree un trabajo de importación de conjuntos de datos y elija Publicar métricas para este trabajo de importación en Publicar métricas de eventos en S3.

Para obtener instrucciones paso a paso, consulte Creación de un trabajo de exportación de conjuntos de datos (consola).

Publicación de métricas de datos masivos en Amazon S3 (AWS CLI)

Para publicar métricas en un bucket de Amazon S3 con AWS Command Line Interface (AWS CLI), utilice el siguiente código para crear un trabajo de importación de conjuntos de datos y proporcionar la `publishAttributionMetricsToS3` marca. Si no desea publicar métricas para un trabajo en concreto, omita la marca. Para obtener información acerca de cada parámetro, consulte 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 INCREMENTAL \
--publish-attribution-metrics-to-s3
```

Publicación de métricas de datos masivos en Amazon S3 (AWS SDK)

Para publicar métricas en un bucket de Amazon S3 con los AWS SDK, cree un trabajo de importación de conjuntos de datos y establézcalo en `publishAttributionMetricsToS3` true. Para obtener información acerca de cada parámetro, consulte CreateDatasetImportJob.

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();
```

Medir el impacto de las recomendaciones mediante A/B pruebas

La realización de una A/B prueba consiste en realizar un experimento con múltiples variaciones y comparar los resultados. Realizar A/B pruebas con las recomendaciones de Amazon Personalize implica mostrar diferentes tipos de recomendaciones a distintos grupos de usuarios y, a continuación, comparar los resultados. Puede utilizar A/B las pruebas para comparar y evaluar diferentes estrategias de recomendación y medir el impacto de las recomendaciones.

Por ejemplo, puedes A/B realizar pruebas para comprobar si las recomendaciones de Amazon Personalize aumentan el porcentaje de clics. Para probar este escenario, puede mostrar recomendaciones a un grupo de usuarios que no estén personalizadas, como los productos destacados. Y puede mostrar a otro grupo recomendaciones personalizadas generadas por Amazon Personalize. A medida que sus clientes interactúan con los elementos, puede registrar los resultados y ver qué estrategia logra la mayor tasa de clics.

El flujo de trabajo para realizar A/B las pruebas con las recomendaciones de Amazon Personalize es el siguiente:

1. Planifique el experimento: defina una hipótesis cuantificable, identifique los objetivos empresariales, defina las variaciones del experimento y determine el marco temporal del experimento.
2. Divida a los usuarios: divida a los usuarios en dos o más grupos, con un grupo de control y uno o más grupos de experimentos.
3. Ejecute su experimento: muestre a los usuarios del grupo de experimentos las recomendaciones modificadas. Muestra las recomendaciones sin cambios a los usuarios del grupo de control. Registre sus interacciones con recomendaciones para realizar un seguimiento de los resultados.
4. Evalúe los resultados: analice los resultados del experimento para determinar si la modificación supuso una diferencia estadísticamente significativa para el grupo del experimento.

Temas

- Prácticas recomendadas para pruebas A/B

Prácticas recomendadas para pruebas A/B

Usa las siguientes prácticas recomendadas para ayudarte a diseñar y mantener A/B las pruebas de las recomendaciones de Amazon Personalize.

- Identifique un objetivo empresarial cuantificable. Compruebe que las diferentes recomendaciones que desea comparar están en consonancia con este objetivo empresarial y no están relacionadas con objetivos diferentes o no cuantificables.
- Defina una hipótesis cuantificable que esté en consonancia con su objetivo empresarial. Por ejemplo, puede predecir que una promoción de su propio contenido personalizado generará un 20 % más de clics en estos elementos. Su hipótesis determina la modificación que realice para su grupo del experimento.
- Defina los indicadores clave de rendimiento relevantes (KPIs) relacionados con su hipótesis. Se utiliza KPIs para medir el resultado de los experimentos. Estos pueden ser los siguientes:
  - Tasa de clics
  - Tiempo de visualización
  - Precio total
- Compruebe que el número total de usuarios del experimento es lo suficientemente grande como para alcanzar un resultado estadísticamente significativo, en función de su hipótesis.
- Defina su estrategia de división del tráfico antes de empezar el experimento. Evite cambiar la división del tráfico mientras se esté llevando a cabo el experimento.
- Mantenga la misma experiencia del usuario de su aplicación o sitio web tanto para el grupo del experimento como para el grupo de control, excepto en lo que respecta a las modificaciones relacionadas con el experimento (por ejemplo, el modelo). Las variaciones en la experiencia del usuario, como la IU o la latencia, pueden generar resultados engañosos.
- Controle los factores externos, como los días festivos, las campañas de marketing en curso y las limitaciones del navegador. Estos factores externos pueden llevar a resultados engañosos.
- Evite cambiar las recomendaciones de Amazon Personalize a menos que estén directamente relacionadas con su hipótesis o requisitos empresariales. Los cambios, como la aplicación de filtros o el cambio manual del orden, pueden generar resultados engañosos.
- Cuando evalúe los resultados, asegúrese de que sean estadísticamente significativos antes de sacar conclusiones. El estándar del sector es un nivel de significancia del 5 %.

Personalización de los resultados de búsqueda desde OpenSearch

Puede utilizar Amazon Personalize para personalizar los resultados de OpenSearch de código abierto o Amazon OpenSearch Service para sus usuarios.

OpenSearch es un servicio de búsqueda autogestionado de código abierto basado en la licencia Apache 2.0. Amazon OpenSearch Service es un servicio administrado que facilita la implementación, la operación y el escalado de recursos de OpenSearch en la nube de AWS. Cuando utiliza Amazon OpenSearch Service, OpenSearch recupera y clasifica los resultados.

Al clasificar los resultados, OpenSearch utiliza un marco de clasificación probabilístico llamado BM-25 para calcular las puntuaciones de relevancia. Si una palabra clave distintiva aparece con más frecuencia en un documento, BM-25 asigna una puntuación de relevancia más alta a ese documento. La clasificación de OpenSearch no tiene en cuenta el comportamiento de los usuarios, como los datos de clics.

Cuando utiliza Amazon Personalize con OpenSearch, Amazon Personalize cambia el orden de los resultados de OpenSearch en función del comportamiento anterior del usuario, los metadatos sobre los elementos y los metadatos sobre el usuario. Luego, OpenSearch incorpora la reclasificación antes de devolver la respuesta de búsqueda a su aplicación. Es usted quien controla el peso que OpenSearch da a la clasificación de Amazon Personalize al aplicarla a los resultados de OpenSearch.

Con esta nueva clasificación, los resultados pueden ser más atractivos y relevantes para los intereses del usuario. Esto puede provocar un aumento en la tasa de clics y la tasa de conversión de su aplicación. Para ver un ejemplo de caso de uso que describe cómo la búsqueda personalizada puede mejorar los resultados de una aplicación de comercio electrónico, consulte Ejemplo de casos de uso.

Antes de empezar a personalizar los resultados de OpenSearch, revise los requisitos que se indican en Requisitos del complemento Search Ranking de Amazon Personalize.

Temas

- Ejemplo de casos de uso
- Funcionamiento del complemento Search Ranking de Amazon Personalize
- Información adicional

- Requisitos del complemento Search Ranking de Amazon Personalize
- Personalización de los resultados de Amazon OpenSearch Service con Amazon Personalize
- Personalización de los resultados de Open Search de código abierto con Amazon Personalize
- Campos para el procesador de respuestas personalized_search_ranking
- Ejemplo de métricas de canalización

Ejemplo de casos de uso

Cuando utiliza Amazon Personalize para volver a clasificar los resultados de OpenSearch, los resultados de la búsqueda pueden ser más relevantes para sus usuarios. Por ejemplo, es posible que tenga una aplicación de comercio electrónico que vende automóviles. Si el usuario introduce una consulta sobre automóviles Toyota y usted no personaliza los resultados, OpenSearch mostrará una lista de automóviles fabricados por Toyota basada en las palabras clave de sus datos. Esta lista se clasificaría en el mismo orden para todos los usuarios.

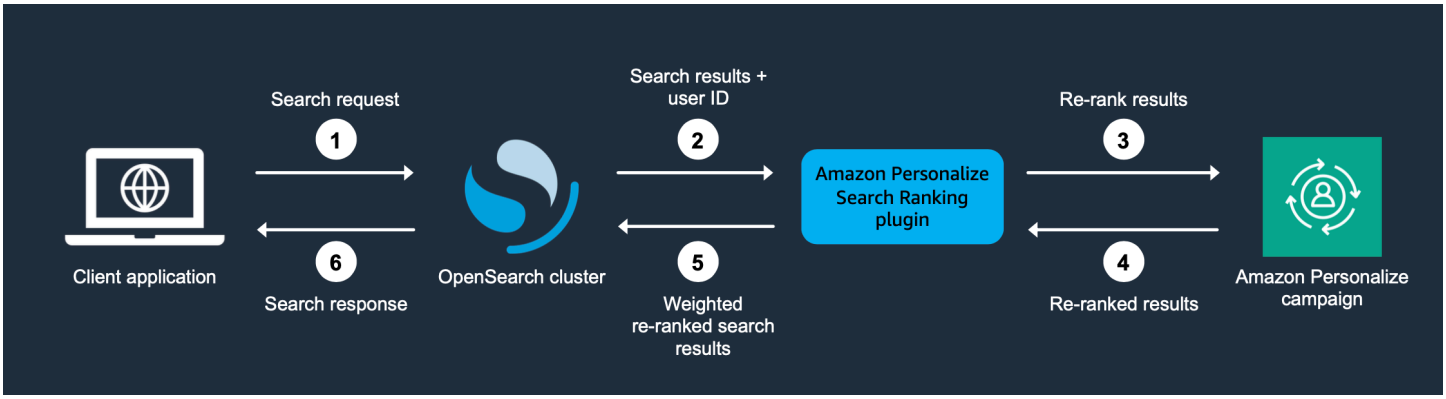
Sin embargo, si utiliza Amazon Personalize para personalizar los resultados, OpenSearch vuelve a clasificar estos coches en orden de relevancia para el usuario específico en función de su comportamiento, por ejemplo, sus clics. El coche en el que es más probable que el usuario haga clic aparece en primer lugar.

Cuando personaliza los resultados de OpenSearch, controla el peso (énfasis) que OpenSearch otorga a la clasificación de Amazon Personalize. Siguiendo con este ejemplo, si un usuario busca un tipo específico de automóvil de un año específico (como un Toyota Prius de 2008), es posible que desee poner más énfasis en la clasificación original de OpenSearch.

Sin embargo, en el caso de consultas más genéricas que generen una amplia gama de resultados (como una búsqueda de todos los vehículos Toyota), podría hacer mucho hincapié en la personalización. De esta forma, los vehículos de la parte superior de la lista son más relevantes para ese usuario en particular.

Funcionamiento del complemento Search Ranking de Amazon Personalize

En el siguiente diagrama se muestra cómo funciona el complemento Search Ranking de Amazon Personalize.



1. Envía la consulta de su cliente a su dominio de OpenSearch Service o a su clúster de OpenSearch de código abierto.
2. OpenSearch envía la respuesta a la consulta (lista de elementos relevantes para la consulta) y el ID del usuario al complemento Search Ranking de Amazon Personalize.
3. El complemento envía los elementos y el usuario en respuesta a su campaña de Amazon Personalize para clasificarlos. Utiliza los valores del nombre de recurso de Amazon (ARN) de la receta y la campaña en su canalización de búsqueda para obtener una clasificación personalizada para el usuario. Utiliza la operación de la API GetPersonalizedRanking para hacer recomendaciones. En la solicitud, pasa el userId del usuario que realiza la consulta y los elementos devueltos por la consulta de OpenSearch en `inputList`.
4. Amazon Personalize devuelve los resultados reclasificados al complemento.
5. El complemento reorganiza y devuelve los resultados de la búsqueda a su dominio de OpenSearch Service o al clúster de OpenSearch de código abierto. Cambia la clasificación de los resultados en función de la respuesta de su campaña de Amazon Personalize y del énfasis en la personalización que especifique durante la configuración.
6. Su clúster OpenSearch de código abierto o su dominio OpenSearch Service devuelve los resultados finales a su aplicación.

Información adicional

En los siguientes recursos se proporciona información adicional acerca del uso de OpenSearch.

- Para obtener más información acerca de cómo comenzar a usar OpenSearch de código abierto, consulte Inicio rápido.
- Para obtener información sobre cómo empezar a usar OpenSearch Service, consulte Introducción a Amazon OpenSearch Service en la Guía para desarrolladores de Amazon OpenSearch Service.

- Para obtener información sobre las recetas Personalized-Ranking en Amazon Personalize, consulte Receta Personalized-Ranking-v2 o Receta Personalized-Ranking.

Requisitos del complemento Search Ranking de Amazon Personalize

Antes de empezar a personalizar los resultados de OpenSearch, tenga en cuenta las siguientes directrices y requisitos para el complemento Search Ranking de Amazon Personalize:

- Debe usar la versión 2.9.0 o una posterior de OpenSearch. Si utiliza Amazon OpenSearch Service, su dominio debe usar la versión 2.9 o posterior.
- Si aún no lo ha hecho, complete las instrucciones de Configuración de permisos para conceder a sus usuarios permiso para acceder a Amazon Personalize y a Amazon Personalize para acceder a sus recursos en Amazon Personalize.
- Debe poder acceder a los recursos de Amazon Personalize desde su dominio de OpenSearch Service o desde un clúster de OpenSearch de código abierto.
  - Para obtener información sobre la concesión de acceso a un dominio de OpenSearch Service, consulte Configuración de los permisos OpenSearch de Amazon Service.
  - Para obtener información sobre la concesión de acceso a un clúster de OpenSearch, consulte Configuración de OpenSearch permisos de código abierto.
- Solo puede usar recursos personalizados de Amazon Personalize. Si creó un grupo de conjuntos de datos de dominio, aún puede agregar recursos personalizados.
- Solo puede usar las recetas personalizadas Receta Personalized-Ranking-v2 o Receta Personalized-Ranking.
- Debe crear un conjunto de datos de interacciones de elementos en Amazon Personalize. Los conjuntos de datos de elementos y usuarios son opcionales.
- No puede aplicar filtros de Amazon Personalize cuando utilice el complemento Search Ranking de Amazon Personalize.
- De forma predeterminada, el complemento asume que el valor `_id` de un documento indexado en OpenSearch coincide con el itemId de sus datos de Amazon Personalize. Si sus datos de OpenSearch utilizan un campo diferente que se corresponde con sus itemIds de Amazon Personalize, debe especificar el nombre del campo al configurar el complemento.
- El userId que utilice para un usuario que realice una consulta debe coincidir con su userId en los datos que importe a Amazon Personalize.

- El complemento vuelve a clasificar solo los 500 resultados de búsqueda principales de OpenSearch. Los elementos restantes no se reclasifican y terminan en la parte inferior de la lista.

Personalización de los resultados de Amazon OpenSearch Service con Amazon Personalize

Para personalizar OpenSearch los resultados de Amazon OpenSearch Service, haz lo siguiente:

1. Configure Amazon Personalize: si aún no lo ha hecho, complete los pasos de Configuración de Amazon Personalize para configurar sus credenciales y configurar los permisos para Amazon Personalize. No necesitas configurar el AWS SDKs para personalizar OpenSearch los resultados.
2. Complete el flujo de trabajo de Amazon Personalize: complete el flujo de trabajo de Amazon Personalize para importar datos, crear una solución con la fórmula Personalized-Ranking-v 2 o la receta de clasificación personalizada, preparar una versión de solución personalizada e implementarla en una campaña. Debe crear un conjunto de datos de interacciones de elementos. Un conjunto de datos de elementos y usuarios son opcionales. Para obtener más información, consulte Detalles del flujo de trabajo de Amazon Personalize.
3. Configurar permisos: configure los permisos para poder acceder a los recursos de Amazon Personalize desde su dominio de OpenSearch servicio. Para obtener más información, consulte Configuración de permisos.
4. Instale el complemento Search Ranking de Amazon Personalize: este complemento gestiona la comunicación con Amazon Personalize y la reclasificación de los resultados. Para obtener información sobre la instalación del complemento en un dominio OpenSearch de servicio, consulteInstalación del complemento.
5. Configure el complemento Search Ranking de Amazon Personalize: para configurar el complemento, debe crear canalizaciones de búsqueda. Las canalizaciones de búsqueda son conjuntos de procesadores de solicitudes y respuestas. Cuando crea una canalización para el complemento, especifica sus recursos de Amazon Personalize en un procesador de respuestas `personalized_search_ranking`. También puede configurar el peso que el complemento otorga a los resultados de Amazon Personalize cuando cambia la clasificación de los resultados. Para obtener más información, consulte Creación de una canalización.
6. Aplique el complemento Amazon Personalize Search Ranking a OpenSearch las consultas: después de configurar un proceso de búsqueda con un procesador de `personalized_search_ranking` respuestas, estará listo para aplicar el complemento Amazon Personalize Search Ranking a sus OpenSearch consultas y ver los resultados reclasificados. Para

obtener información sobre cómo aplicar el complemento a las consultas OpenSearch del Servicio, consulteAplicación del complemento.

7. Comparar resultados: el complemento Amazon Personalize Search Ranking reclasifica los resultados de búsqueda en la respuesta a la OpenSearch consulta. Considera tanto la clasificación de Amazon Personalize como la clasificación de OpenSearch. Para entender cómo se reclasifican los resultados, puede comparar los resultados de las consultas que utilizan la personalización y las que no. Para obtener información sobre cómo comparar los resultados con los OpenSearch del Servicio, consulteComparación de resultados.
8. Supervise el complemento Search Ranking de Amazon Personalize: al aplicar el complemento Search Ranking de Amazon Personalize a las consultas de búsqueda, puede supervisar el complemento obteniendo métricas de sus canalizaciones de búsqueda. Para obtener información sobre la supervisión del complemento con el OpenSearch Servicio, consulteSupervisión del complemento.

Temas

- Configuración de los permisos OpenSearch de Amazon Service
- Instalación del complemento Search Ranking de Amazon Personalize en un dominio de OpenSearch Service
- Creación de una canalización en Amazon OpenSearch Service
- Aplicación del complemento a las consultas de Amazon OpenSearch Service
- Comparación de los resultados de Amazon OpenSearch Service personalizados con los resultados sin personalización
- Supervisión del complemento con Amazon OpenSearch Service

Configuración de los permisos OpenSearch de Amazon Service

Si utilizas Amazon OpenSearch Service, debes poder acceder a tus recursos de Amazon Personalize desde tu dominio de OpenSearch servicio.

Configuración de permisos de

1. En función de si los recursos están en la misma cuenta o en cuentas diferentes, cree uno o más roles de servicio de IAM con permiso para acceder a los recursos.
   - Si sus recursos de OpenSearch Service y Amazon Personalize están en la misma cuenta, crea un rol de servicio de IAM para OpenSearch Service y le concede permiso para obtener

una clasificación personalizada de su campaña de Amazon Personalize. Para obtener más información, consulte Configuración de permisos cuando los recursos están en la misma cuenta.

- Si sus recursos OpenSearch de Service y Amazon Personalize están en cuentas distintas, debe crear dos funciones de servicio de IAM. Usted crea uno en la cuenta con sus recursos de OpenSearch servicio y le otorga acceso a dichos OpenSearch recursos. Y cree otro en la cuenta con los recursos de Amazon Personalize y concédale permiso para obtener una clasificación personalizada de la campaña de Amazon Personalize. Para obtener más información, consulte Configuración de permisos cuando los recursos están en cuentas distintas.

2. Concede al usuario o rol que accede a tu dominio de OpenSearch servicio `PassRole` permisos para el rol de servicio de IAM que creaste para el OpenSearch Servicio. Para obtener más información, consulte Configuración de la seguridad OpenSearch del dominio de Amazon Service.

Después de configurar los permisos, podrá instalar el complemento en el dominio. Para obtener más información, consulte Instalación del complemento.

Temas

- Configuración de permisos cuando los recursos están en la misma cuenta
- Configuración de permisos cuando los recursos están en cuentas distintas
- Configuración de la seguridad OpenSearch del dominio de Amazon Service

Configuración de permisos cuando los recursos están en la misma cuenta

Si sus recursos de OpenSearch Service y Amazon Personalize están en la misma cuenta, debe crear un rol de servicio de IAM para OpenSearch Service. Este rol debe tener permiso para obtener una clasificación personalizada de su campaña de Amazon Personalize. Para conceder a su función de servicio de OpenSearch servicio permiso para obtener una clasificación personalizada de su campaña Amazon Personalize, haga lo siguiente:

- La política de confianza del rol debe conceder `AssumeRole` permisos para el OpenSearch Servicio. Para ver una política de ejemplo, consulte Política de confianza de ejemplo.
- El rol debe tener permiso para obtener una clasificación personalizada de su campaña de Amazon Personalize. Para ver una política de ejemplo, consulte Ejemplo de política de permisos.

Para obtener información sobre la creación de un rol de IAM, consulte Creación de roles de IAM en la Guía del usuario de IAM. Para obtener información sobre cómo asociar una política de IAM a rol, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

Después de crear un rol de servicio de IAM para el OpenSearch servicio, debes conceder al usuario o rol que accede a tu dominio de OpenSearch servicio `PassRole` permisos para el rol de OpenSearch servicio de servicio. Para obtener más información, consulte Configuración de la seguridad OpenSearch del dominio de Amazon Service.

Temas

- Política de confianza de ejemplo
- Ejemplo de política de permisos

Política de confianza de ejemplo

El siguiente ejemplo de política de confianza concede `AssumeRole` permisos para el OpenSearch Servicio.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Principal": {
            "Service": [
                "es.amazonaws.com"
            ]
        }
    }]
}
```

Ejemplo de política de permisos

El siguiente ejemplo de política concede al rol los permisos mínimos para obtener una clasificación personalizada de su campaña de Amazon Personalize. Para `Campaign ARN`, especifique el nombre de recurso de Amazon (ARN) de la campaña de Amazon Personalize.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

Configuración de permisos cuando los recursos están en cuentas distintas

Si sus recursos de OpenSearch Service y Amazon Personalize están en cuentas distintas, debe crear un rol de IAM en cada cuenta y conceder al rol acceso a los recursos de la cuenta.

Configuración de permisos para varias cuentas

1. En la cuenta en la que se encuentra la campaña de Amazon Personalize, cree un rol de IAM con permiso para obtener una clasificación personalizada de la campaña de Amazon Personalize. Al configurar el complemento, especifique el ARN de este rol en el parámetro `external_account_iam_role_arn` del procesador de respuestas `personalized_search_ranking`. Para obtener más información, consulte Creación de una canalización en Amazon OpenSearch Service.

   Para ver una política de ejemplo, consulte Ejemplo de política de permisos.

2. En la cuenta en la que se encuentre su dominio de OpenSearch servicio, cree un rol con una política de confianza que otorgue `AssumeRole` permisos OpenSearch de servicio. Al configurar el complemento, especifique el ARN de este rol en el parámetro `iam_role_arn` del procesador de respuestas `personalized_search_ranking`. Para obtener más información, consulte Creación de una canalización en Amazon OpenSearch Service.

   Para ver una política de ejemplo, consulte Política de confianza de ejemplo.

3. Modifique cada rol para conceder `AssumeRole` permisos al otro rol. Por ejemplo, para el rol que tiene acceso a sus recursos de Amazon Personalize, su política de IAM otorgaría al rol en la cuenta con el dominio de OpenSearch servicio los permisos de rol de la siguiente manera:

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [{
           "Sid": "",
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::111122223333:role/roleName"

       }]
   }
   ```

4. En la cuenta en la que se encuentre su dominio de OpenSearch servicio, otorgue al usuario o rol que accede a su dominio de OpenSearch servicio `PassRole` permisos para el rol de OpenSearch servicio que acaba de crear. Para obtener más información, consulte Configuración de la seguridad OpenSearch del dominio de Amazon Service.

Configuración de la seguridad OpenSearch del dominio de Amazon Service

Para usar el complemento con el OpenSearch Servicio, el usuario o rol que accede a tu dominio debe tener `PassRole` permisos para el rol de servicio de IAM del OpenSearch Servicio que acabas de crear. Además, el usuario o rol deben tener permiso para realizar las acciones `es:ESHttpGet` y `es:ESHttpPut`.

Para obtener información sobre cómo configurar el acceso al OpenSearch Servicio, consulte Seguridad en Amazon OpenSearch Service en la Guía para desarrolladores de Amazon OpenSearch Service. Para ver ejemplos de políticas, consulte Ejemplos de políticas para el usuario o rol del OpenSearch servicio.

Ejemplos de políticas para el usuario o rol del OpenSearch servicio

El siguiente ejemplo de política de IAM concede a un usuario o rol `PassRole` permisos para el rol de servicio de IAM que creó para OpenSearch Service en el que. Configuración de permisos cuando los recursos están en la misma cuenta

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 siguiente política de IAM concede los permisos mínimos para crear canalizaciones y consultas de búsqueda con Service. OpenSearch

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:ESHttpGet",
                "es:ESHttpPut"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:ResourceTag/environment": [
                        "production"
                    ]
                }
            }
        }
    ]
}
```

Instalación del complemento Search Ranking de Amazon Personalize en un dominio de OpenSearch Service

Tras completar el flujo de trabajo de Amazon Personalize y cumplir los requisitos que se indican en Requisitos del complemento, estará listo para instalar el complemento en el dominio.

Para usar el complemento, debe asociar `Amazon_Personalize_Search_Ranking_Plugin` al dominio. El complemento está preinstalado y no es necesario importarlo desde Amazon S3. El complemento se asocia de la misma manera que se asocia un paquete de OpenSearch Service. Para obtener información sobre cómo asociar un paquete de OpenSearch Service, consulte Paquetes personalizados para Amazon OpenSearch Service.

Después de asociar el complemento al dominio, estará listo para configurarlo. Para configurarlo, cree una canalización de búsqueda y especifique un procesador de respuestas `personalized_search_ranking`. Para obtener más información, consulte Creación de una canalización.

Información adicional sobre dominios de Amazon OpenSearch Service

En los siguientes recursos se proporciona información adicional sobre el uso del dominio de Amazon OpenSearch Service.

- Para ver un tutorial conciso sobre cómo configurar un dominio de prueba, consulte el paso 1: Crear un dominio de Amazon OpenSearch Service en la sección “Introducción” de la Guía para desarrolladores de Amazon OpenSearch Service.
- Para obtener pasos más detallados sobre la configuración de los dominios de OpenSearch Service, consulte Creación y administración de dominios de Amazon OpenSearch Service.
- Para ver un tutorial conciso sobre cómo subir una pequeña cantidad de datos de prueba en OpenSearch Service, consulte el Paso 2: Aubir datos a Amazon OpenSearch Service para su indexación en la sección “Introducción” de la Guía para desarrolladores de Amazon OpenSearch Service.
- Para obtener información completa sobre la ingesta de datos, consulte Indexación de datos en Amazon OpenSearch Service en la Guía para desarrolladores de Amazon OpenSearch Service.

Creación de una canalización en Amazon OpenSearch Service

Tras instalar el complemento Search Ranking de Amazon Personalize, estará listo para configurarlo mediante la creación de una canalización de búsqueda de OpenSearch.

Una canalización de búsqueda es un conjunto de procesadores de solicitudes y respuestas que se ejecutan secuencialmente en el orden en que se crean. Cuando crea una canalización de búsqueda para el complemento, especifica un procesador de respuestas `personalized_search_ranking`. Para obtener información sobre las canalizaciones de búsqueda, consulta Buscar canalizaciones.

Tras crear una canalización de búsqueda con un procesador de respuestas `personalized_search_ranking`, estará listo para empezar a aplicar el complemento a las consultas de OpenSearch. Puede aplicarlo a un índice de OpenSearch o a una consulta individual de OpenSearch. Para obtener más información, consulte Aplicación del complemento.

Puede usar el siguiente código de Python para crear una canalización de búsqueda con un procesador de respuestas `personalized_search_ranking` en un dominio de OpenSearch Service. Sustituya `domain endpoint` por la URL del punto de conexión de su dominio. Por ejemplo: `https://<domain name>.<AWS region>.es-staging.amazonaws.com`. Para obtener una explicación completa de cada parámetro de `personalized_search_ranking`, consulte Campos para el procesador de respuestas `personalized_search_ranking`.

```
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}")
```

Aplicación del complemento a las consultas de Amazon OpenSearch Service

Después de crear un canalización, podrá aplicar el complemento Search Ranking de Amazon Personalize a las consultas. Puede aplicar el complemento Search Ranking de Amazon Personalize a todas las consultas y respuestas de un índice. También puede aplicar el complemento a consultas y respuestas individuales.

- Puede usar el siguiente código de Python para aplicar una canalización de búsqueda a un índice. Con este enfoque, todas las búsquedas que utilizan este índice utilizan el complemento para personalizar los resultados de la búsqueda.

```
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}")
```

- Puede usar el siguiente código de Python para aplicar una canalización de búsqueda a una consulta individual para los coches de la marca Toyota.

  Actualice el código para especificar el punto de conexión de su dominio, su índice de OpenSearch Service, el nombre de su canalización y su consulta. Para `user_id`, especifique el ID del usuario del que va a obtener los resultados de la búsqueda. Este usuario debe estar en los datos que utilizó para crear la versión de solución de Amazon Personalize. Si el usuario no estuviera presente, Amazon Personalize clasifica los elementos según su popularidad.

  Para `context`, si utiliza metadatos contextuales, proporcione los metadatos contextuales del usuario, como su tipo de dispositivo. El campo `context` es opcional. Para obtener más información, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales.

```
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}")
```

Comparación de los resultados de Amazon OpenSearch Service personalizados con los resultados sin personalización

El complemento Search Ranking de Amazon Personalize reorganiza los resultados de búsqueda en función de la clasificación de Amazon Personalize y la clasificación de OpenSearch. La forma en que el complemento reclasifica los resultados depende de cómo haya configurado el procesador de respuestas `personalized_search_ranking` en sus canalizaciones.

Para comprender cómo se clasifican los resultados, puede ejecutar consultas con y sin personalización y compararlos. Puede usar el siguiente código de Python para ejecutar dos consultas diferentes y generar los resultados en dos archivos JSON. El primer método ejecuta una consulta que utiliza el complemento para volver a clasificar los resultados. El segundo ejecuta un método que genera resultados sin personalización.

```
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)
```

Supervisión del complemento con Amazon OpenSearch Service

Al aplicar el complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch, puede supervisar el complemento obteniendo métricas para sus procesos de búsqueda. Las métricas de canalización incluyen estadísticas como el número de solicitudes con error del procesador de respuestas `personalized_search_ranking`.

Si utiliza OpenSearch Service, puede supervisar el complemento mediante métricas en Amazon CloudWatch. Para obtener más información, consulte Monitoreo de los dominios de Amazon OpenSearch Service.

Puede utilizar el siguiente código de Python para obtener métricas para todas las canalizaciones. Para obtener un ejemplo de unas métricas de canalización, consulte Ejemplo de métricas de canalización.

```
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}")
```

Personalización de los resultados de Open Search de código abierto con Amazon Personalize

Para personalizar los resultados en código abierto OpenSearch, haga lo siguiente:

1. Configure Amazon Personalize: si aún no lo ha hecho, complete los pasos de Configuración de Amazon Personalize para configurar sus credenciales y configurar los permisos para Amazon Personalize. No es necesario configurar el AWS SDKs para personalizar OpenSearch los resultados.
2. Complete el flujo de trabajo de Amazon Personalize: complete el flujo de trabajo de Amazon Personalize para importar datos, crear una solución con la Personalized-Ranking-v receta de clasificación doble o personalizada, preparar una versión de solución personalizada e implementarla en una campaña. Debe crear un conjunto de datos de interacciones de elementos. Un conjunto de datos de elementos y usuarios son opcionales. Para obtener más información, consulte Detalles del flujo de trabajo de Amazon Personalize.
3. Configurar permisos: configure los permisos para poder acceder a los recursos de Amazon Personalize desde su OpenSearch clúster. Para obtener más información, consulte Configuración de permisos.
4. Instale el complemento Search Ranking de Amazon Personalize: este complemento gestiona la comunicación con Amazon Personalize y la reclasificación de los resultados.
   - Si ya tiene un OpenSearch clúster en ejecución, puede instalar el complemento manualmente. Para obtener más información, consulte Instalación manual del complemento en un clúster de OpenSearch existente.
   - Si no has creado un OpenSearch clúster, puedes usar un script bash de inicio rápido para crear uno. Para obtener más información, consulte Creación de un clúster nuevo e instalación del complemento con un script.
5. Configure el complemento Search Ranking de Amazon Personalize: para configurar el complemento, debe crear canalizaciones de búsqueda. Las canalizaciones de búsqueda son conjuntos de procesadores de solicitudes y respuestas. Cuando crea una canalización para el

complemento, especifica sus recursos de Amazon Personalize en un procesador de respuestas `personalized_search_ranking`. También puede configurar el peso que el complemento otorga a los resultados de Amazon Personalize cuando cambia la clasificación de los resultados. Para obtener más información, consulte Creación de una canalización.

6. Aplicar el complemento Amazon Personalize Search Ranking a OpenSearch las consultas: puede aplicar el complemento Amazon Personalize Search Ranking a todas las consultas y respuestas de un OpenSearch índice. También puedes aplicar el plugin a OpenSearch consultas y respuestas individuales. Para obtener información sobre cómo aplicar el complemento a las consultas de código abierto OpenSearch, consulteAplicación del complemento.
7. Comparar resultados: el complemento Amazon Personalize Search Ranking reclasifica los resultados de búsqueda en la respuesta a la OpenSearch consulta. Considera tanto la clasificación de Amazon Personalize como la clasificación de OpenSearch. Para entender cómo se reclasifican los resultados, puede comparar los resultados de las consultas que utilizan la personalización y las que no. Para obtener información sobre cómo comparar los resultados con los de código abierto OpenSearch, consulteComparación de resultados.
8. Supervise el complemento Search Ranking de Amazon Personalize: al aplicar el complemento Search Ranking de Amazon Personalize a las consultas de búsqueda, puede supervisar el complemento obteniendo métricas de sus canalizaciones de búsqueda. Para obtener información sobre la supervisión del complemento en un OpenSearch clúster de código abierto, consulteSupervisión del complemento con OpenSearch de código abierto. Para ver un extracto de las métricas de canalización devueltas OpenSearch, consulteEjemplo de métricas de canalización.

Temas

- Configuración de OpenSearch permisos de código abierto
- Instalación manual del complemento Search Ranking de Amazon Personalize en un clúster de OpenSearch existente
- Creación de un clúster nuevo e instalación del complemento con un script
- Creación de una canalización en OpenSearch de código abierto
- Aplicación del complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch de código abierto
- Comparación de los resultados de OpenSearch personalizados con los resultados sin personalización
- Supervisión del complemento con OpenSearch de código abierto

Configuración de OpenSearch permisos de código abierto

Si utilizas código abierto OpenSearch, debes poder acceder a tus recursos de Amazon Personalize desde tu clúster de búsqueda abierto. Para otorgar acceso, realice lo siguiente:

- Si está configurando OpenSearch desde cero, puede usar un script bash de inicio rápido para ejecutar un OpenSearch clúster en un contenedor de Docker. El script usa las credenciales predeterminadas de tu perfil. AWS Puede especificar un perfil alternativo al ejecutar el script.

  Estas credenciales deben estar asociadas a un usuario o rol que tenga permiso para realizar la GetPersonalizedRanking acción de tu campaña Amazon Personalize. Para ver una política de IAM de ejemplo, consulte Ejemplos de políticas de IAM. Como alternativa, las credenciales deben tener permiso para asumir un rol que tenga estos permisos. Puede proporcionar el nombre de recurso de Amazon (ARN) para este rol al crear un canalización para el complemento Search Ranking de Amazon Personalize.
- Si no utilizas el script bash de inicio rápido, puedes añadir tus credenciales manualmente a tu almacén de OpenSearch claves. Estas credenciales deben corresponder a un usuario o rol que tenga permiso para realizar la GetPersonalizedRanking acción de tu campaña Amazon Personalize.

  Para añadir tus AWS credenciales manualmente al OpenSearch almacén de claves, ejecuta el siguiente comando en el lugar en el que se ejecute el OpenSearch clúster (por ejemplo, en un contenedor de Docker). A continuación, proporcione cada credencial. Si no utiliza un token de sesión, puede omitir la última línea del comando.

  ```
  opensearch-keystore add \
  personalized_search_ranking.aws.access_key \
  personalized_search_ranking.aws.secret_key \
  personalized_search_ranking.aws.session_token
  ```
- Si ejecuta el OpenSearch clúster en una instancia de Amazon EC2, puede conceder permisos con un perfil de instancia de IAM. La política asociada al rol debe concederle permiso para realizar la GetPersonalizedRanking acción en tu campaña Amazon Personalize. También conceder el permisos de Amazon EC2 para asumir el rol.

  Para obtener más información sobre los perfiles de instancia de Amazon EC2, consulte Uso de perfiles de instancia. Para ver una política de ejemplo, consulte Ejemplos de políticas de IAM.

Ejemplos de políticas de IAM

El siguiente ejemplo de política concede a un usuario o rol los permisos mínimos para obtener una clasificación personalizada de su campaña de Amazon Personalize. Para `Campaign ARN`, especifique el nombre de recurso de Amazon (ARN) de la campaña de Amazon Personalize.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

Además, si ejecuta el OpenSearch clúster en una instancia de Amazon EC2 y concede permisos con un perfil de instancia de IAM, la política de confianza del rol debe conceder los permisos de Amazon `AssumeRole` EC2 de la siguiente manera. Para obtener más información sobre los perfiles de instancia de Amazon EC2, consulte Uso de perfiles de instancia.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

Instalación manual del complemento Search Ranking de Amazon Personalize en un clúster de OpenSearch existente

Si ya tiene un clúster de OpenSearch, puede instalar manualmente el complemento en su clúster directamente desde el repositorio de GitHub de OpenSearch.

Para instalar el complemento manualmente

1. Utilice el siguiente comando para iniciar el clúster de OpenSearch:

   ```
   bin/opensearch
   ```

2. Si aún no lo ha hecho, suba datos de catálogo a su clúster de OpenSearch. Al subir los datos, se crea un índice de OpenSearch y se definen las asignaciones de campos. A continuación suba sus datos a ese índice. Para ver un ejemplo, consulte Crear un índice y asignaciones de campos con datos de muestra.
3. Utilice el siguiente comando para instalar el complemento.

   ```
   bin/opensearch-plugin install https://github.com/opensearch-project/search-processor/releases/download/2.9.0/opensearch-search-processor-2.9.0.0.zip
   ```

   Para obtener más información sobre la instalación de complementos, consulte Instalación de complementos.

Después de instalar el complemento Search Ranking de Amazon Personalize, estará listo para configurarlo. Para configurar el complemento, cree una canalización de búsqueda y especifique un procesador de respuestas `personalized_search_ranking`. Para obtener más información, consulte Creación de una canalización.

Creación de un clúster nuevo e instalación del complemento con un script

Si no ha creado un clúster de OpenSearch, puede utilizar un script bash de inicio rápido para crear uno. Este script configura un clúster de OpenSearch en un contenedor de Docker, configura

las credenciales con su perfil AWS predeterminado e instala el complemento Search Ranking de Amazon Personalize.

Para obtener información sobre cómo crear manualmente un clúster de OpenSearch, consulte las instrucciones de inicio rápido en la documentación de OpenSearch.

Para instalar el complemento con un script bash de inicio rápido

1. Antes de ejecutar el script, descargue e instale Docker de escritorio para su sistema operativo.
2. Descargue el script bash de inicio rápido desde GitHub.
3. En el directorio de trabajo, ejecute el script con el siguiente comando.

   ```
   sh personalized_search_ranking_quickstart.sh
   ```

   Con este comando, el script usa las credenciales en su perfil AWS predeterminado. Para proporcionar un perfil alternativo, utilice el argumento `--profile`.

   ```
   sh personalized_search_ranking_quickstart.sh --profile profile-name
   ```

   Tras ejecutar el script, encontrará más información sobre el script en el archivo README que se encuentra en el directorio único creado por el script. Este directorio almacena los archivos Dockerfile y docker-compose.yml que utiliza el script. Por ejemplo: `../opensearch-personalize-intelligent-ranking-docker.1234/README`.
4. Suba datos de catálogo a su clúster de OpenSearch. Al subir los datos, se crea un índice de OpenSearch y se definen las asignaciones de campos. A continuación suba sus datos a ese índice. Para ver un ejemplo, consulte Crear un índice y asignaciones de campos con datos de muestra.

Tras configurar OpenSearch e instalar el complemento Search Ranking de Amazon Personalize, estará listo para configurarlo. Para configurar el complemento, cree una canalización de búsqueda y especifique un procesador de respuestas `personalized_search_ranking`. Para obtener más información, consulte Creación de una canalización.

Creación de una canalización en OpenSearch de código abierto

Tras instalar el complemento en el clúster, estará listo para configurarlo mediante la creación de una canalización de búsqueda de OpenSearch.

Una canalización de búsqueda es un conjunto de procesadores de solicitudes y respuestas que se ejecutan secuencialmente en el orden en que se crean. Cuando crea una canalización de búsqueda para el complemento, especifica un procesador de respuestas `personalized_search_ranking`. Para obtener información sobre las canalizaciones de búsqueda, consulta Buscar canalizaciones.

Después de crear una canalización con un procesador de respuestas `personalized_search_ranking`, podrá empezar a aplicar el complemento a las consultas. Para obtener más información, consulte Aplicación del complemento.

Puede utilizar el siguiente comando curl para crear una canalización de búsqueda con un procesador de respuesta `personalized_search_ranking` en un clúster de OpenSearch de código abierto. Para obtener una explicación completa de cada parámetro de `personalized_search_ranking`, consulte Campos para el procesador de respuestas `personalized_search_ranking`.

```
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
      }
    }
  ]
}'
```

Tras crear una canalización de búsqueda con un procesador de respuestas `personalized_search_ranking`, estará listo para empezar a aplicar el complemento a las consultas de OpenSearch. Puede aplicarlo a un índice de OpenSearch o a una consulta individual de OpenSearch. Para obtener más información, consulte Aplicación del complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch de código abierto.

Aplicación del complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch de código abierto

Puede aplicar el complemento Search Ranking de Amazon Personalize a todas las consultas y respuestas de un índice de OpenSearch. También puede aplicar el complemento a consultas y respuestas individuales de OpenSearch.

- El siguiente comando curl aplica una canalización de búsqueda a un índice de OpenSearch en un clúster de OpenSearch de código abierto que se ejecuta localmente. Con este enfoque, todas las búsquedas de este índice utilizan el complemento para personalizar los resultados de la búsqueda.

```
curl -XGET "https://localhost:9200/index/_settings" -ku 'admin:admin' --insecure -H
 'Content-Type: application/json' -d'
{
  "index.search.default_pipeline": "pipeline-name"
}
'
```

- El siguiente comando curl aplica una canalización de búsqueda a una consulta individual para los coches de la marca Toyota a un índice de un clúster de OpenSearch de código abierto que se ejecuta localmente.

  Para `user_id`, especifique el ID del usuario del que va a obtener los resultados de la búsqueda. Este usuario debe estar en los datos que utilizó para crear la versión de solución de Amazon Personalize. Si el usuario no estuviera presente, Amazon Personalize clasifica los elementos según su popularidad. Para `context`, si utiliza metadatos contextuales, proporcione los metadatos contextuales del usuario, como su tipo de dispositivo. El campo `context` es opcional. Para obtener más información, consulte Aumento de la relevancia de las recomendaciones con metadatos contextuales.

```
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" }
    }
  }
}
'
```

Para entender cómo se reclasifican los resultados, puede usar OpenSearch Dashboards para comparar los resultados de OpenSearch con los resultados reclasificados con el complemento. Para obtener más información, consulte Comparación de los resultados de OpenSearch personalizados con los resultados sin personalización.

A medida que vaya aplicando el complemento a las consultas de OpenSearch, podrá supervisarlo obteniendo métricas para su canalización de OpenSearch. Para obtener más información, consulte Supervisión del complemento con OpenSearch de código abierto.

Comparación de los resultados de OpenSearch personalizados con los resultados sin personalización

Para entender cómo se reclasifican los resultados, puede ejecutar consultas con la consola de Dev Tools en dos ventanas de navegador independientes. A continuación, puede comparar los resultados de las consultas con y sin personalización.

Para comparar los resultados con la consola de Dev Tools

1. Asegúrese de que OpenSearch Dashboards esté instalado. El script bash de inicio rápido instala OpenSearch Dashboards. Si no usa el script o ya tiene un clúster en ejecución, debe instalar OpenSearch Dashboards. Para obtener más información, consulte Instalación de OpenSearch Dashboards.
2. Inicie OpenSearch Dashboards. Abra `http://localhost:5601` desde un navegador e inicie sesión en OpenSearch Dashboards. Las credenciales predeterminadas son el nombre de usuario “admin” y la contraseña “admin”.
3. Elija Dev Tools en el menú de administración de la página de inicio de OpenSearch Dashboards.
4. Abra otra ventana del navegador y vuelva a abrir la consola de Dev Tools. Puede usar la URL de la ventana anterior.

5. En una ventana, escriba una consulta que no utilice ninguna reordenación para personalizarla. En la otra ventana, introduzca un comando curl que utilice una canalización con el procesador de respuestas `personalized_search_ranking`. Si pega un comando curl directamente en la consola, el comando se convierte automáticamente al formato que utilice la consola. Para ver un comando de ejemplo, consulte Aplicación del complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch de código abierto.
6. Ejecute las dos consultas y compare los resultados.

Supervisión del complemento con OpenSearch de código abierto

Al aplicar el complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch, puede supervisar el complemento obteniendo métricas para sus procesos de búsqueda. Las métricas de canalización incluyen estadísticas como el número de solicitudes con error del procesador de respuestas `personalized_search_ranking`.

Puede utilizar el siguiente código para obtener métricas para todas las canalizaciones. La respuesta contiene estadísticas de todas las canalizaciones de búsqueda. Para obtener un ejemplo de unas métricas de canalización, consulte Ejemplo de métricas de canalización.

```
curl -XGET "https://localhost:9200/_nodes/stats/search_pipeline?pretty" -ku
 'admin:admin'
```

Campos para el procesador de respuestas **`personalized_search_ranking`**

Cuando crea una canalización de búsqueda para el complemento Search Ranking de Amazon Personalize, especifica un procesador de respuestas `personalized_search_ranking` con los siguientes campos.

- campaign_arn (obligatorio): especifique el nombre del recurso de Amazon (ARN) de la campaña de Amazon Personalize que se utilizará para personalizar los resultados.
- item_id_field (opcional): si el campo `_id` de un documento indexado en OpenSearch no se corresponde con sus itemIds de Amazon Personalize, especifique el nombre del campo que sí lo hace. De forma predeterminada, el complemento asume que los datos `_id` coinciden con el itemId de sus datos de Amazon Personalize.

- recipe (obligatorio): especifica el nombre de la receta de Amazon Personalize que se va a utilizar. Para usar el complemento, puede especificar `aws-personalized-ranking-v2` o `aws-personalized-ranking`.
- weight (obligatorio): especifique el énfasis que el procesador de respuestas pone en la personalización cuando reclasifica los resultados. Especifique un valor dentro de un rango de 0,0 —1,0. Cuanto más cerca esté de `1.0`, más probable es que los resultados de Amazon Personalize tengan una clasificación más alta. Si especifica `0.0`, no se produce ninguna personalización y OpenSearch tiene prioridad.
- tag (opcional): especifique un identificador para el procesador.
- iam_role_arn (obligatorio para OpenSearch Service, opcional para OpenSearch de código abierto): para OpenSearch Service, indique el nombre de recurso de Amazon (ARN) rol que ha creado al configurar los permisos para que OpenSearch Service acceda a los recursos de Amazon Personalize. Si los recursos de OpenSearch Service y Amazon Personalize existen en cuentas diferentes, especifique el rol que concede permisos `AssumeRole` para OpenSearch Service. Para obtener más información, consulte Configuración de permisos cuando los recursos están en cuentas distintas.

  En el caso de OpenSearch de código abierto, si utiliza varios roles para restringir los permisos a distintos grupos de usuarios de la organización, especifique el ARN del rol que tiene permiso para acceder a Amazon Personalize. Si utiliza únicamente las credenciales AWS de su almacén de claves de OpenSearch, puede omitir este campo.
- aws_region (obligatorio): la región AWS en la que creó su campaña de Amazon Personalize.
- ignore_failure (opcional): especifique si el complemento ignora los errores del procesador. Para los valores, especifique `true` o `false`. Para sus entornos de producción, le recomendamos que lo especifique `true` para evitar interrupciones en las respuestas a las consultas. En el caso de los entornos de prueba, puede especificar `false` para ver los errores que genera el complemento.
- external_account_iam_role_arn: si utiliza OpenSearch Service y los recursos de Amazon Personalize y OpenSearch Service están en cuentas diferentes, especifique el ARN del rol que tiene permiso para acceder a los recursos de Amazon Personalize. Este rol debe estar en la misma cuenta que los recursos de Amazon Personalize. Para obtener más información, consulte Configuración de permisos cuando los recursos están en cuentas distintas.

Para ver una muestra de código de OpenSearch Service, consulte Creación de una canalización en Amazon OpenSearch Service. Para ver un ejemplo de OpenSearch de código abierto, consulte Creación de una canalización en OpenSearch de código abierto.

Ejemplo de métricas de canalización

Al aplicar el complemento Search Ranking de Amazon Personalize a las consultas de OpenSearch, puede supervisar el complemento obteniendo métricas para sus procesos de búsqueda. Las métricas de canalización incluyen estadísticas como el número de solicitudes con error del procesador de respuestas `personalized_search_ranking`.

En el siguiente código se muestra un extracto de las métricas de canalización devueltas por OpenSearch. Muestra solo el objeto `pipelines` que contiene las estadísticas de dos canalizaciones diferentes. Para cada canalización, puede encontrar las métricas del complemento Search Ranking de Amazon Personalize en la lista de procesadores de respuestas `personalized_search_ranking`. Para ver un ejemplo completo de todas las métricas, consulte Búsqueda de métricas de canalización.

```
{
....
....
  "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>
            }
          }
        }
      ]
    }
  }
....
....
}
```

Etiquetado de recursos de Amazon Personalize

Una etiqueta es una etiqueta que puede definir y asociar de forma opcional a AWS los recursos, incluidos determinados tipos de recursos de Amazon Personalize. Un recurso puede tener hasta 50 etiquetas.

Las etiquetas pueden ayudarle a clasificar y administrar recursos de distintas formas, como, por finalidad, entorno u otros criterios. Por ejemplo, puede utilizar etiquetas para dividir los ingresos entre diferentes funciones o identificar entornos de desarrollo para recursos diferentes.

Para recuperar los recursos de Amazon Personalize por etiqueta, puede utilizar los filtros en GetResources funcionamiento de la API Resource Groups Tagging. Para obtener más información, consulte la guía GetResourcesde referencia de la API de etiquetado de Resource Groups.

Puede agregar etiquetas a los siguientes tipos de recursos de Amazon Personalize:

- Trabajos de inferencia por lotes
- Trabajos de segmentos por lotes
- Campaigns (Campañas)
- Conjuntos de datos
- Grupos de conjuntos de datos
- Trabajos de importación y exportación de conjuntos de datos
- rastreador de eventos
- Filtros
- Recomendadores
- Soluciones
- Versiones de la solución

Temas

- Directrices y requisitos
- Adición de etiquetas a recursos de Amazon Personalize
- Eliminación de etiquetas de los recursos de Amazon Personalize
- Uso de etiquetas en políticas de IAM

Directrices y requisitos

Cada etiqueta está formada por una clave de etiqueta y un valor de etiqueta opcional, ambos definidos por el usuario. Un clave de etiqueta es una etiqueta general que actúa como una categoría para valores de etiqueta más específicos. Un valor de etiqueta actúa como descriptor de una clave de etiqueta.

Por ejemplo, si tiene dos versiones de un grupo de conjuntos de datos de Amazon Personalize (una para pruebas internas y otra para producción), podría asignar una clave de etiqueta `Environment` a ambos proyectos. El valor de la etiqueta `Environment` podría ser `Test` para una versión del grupo de conjuntos de datos y `Production` para la otra versión.

Se aplican las siguientes restricciones a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8
- Las claves y los valores de la etiqueta pueden contener los siguientes caracteres: A-Z, a-z, 0-9, espacio y _ . : / = + @ – (guion). Este es el conjunto estándar de caracteres disponible en todos los servicios AWS que admiten etiquetas. Algunos servicios admiten símbolos adicionales.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- Para cada recurso asociado, cada clave de etiqueta debe ser única y solo puede tener un valor de etiqueta.
- Las claves y los valores de las etiquetas no pueden empezar por. `aws:` AWS los servicios aplican etiquetas que comienzan por`aws:`, y esas etiquetas no se pueden modificar. No se tienen en cuenta para los límites de etiquetas.
- No puede actualizar ni eliminar un recurso en función únicamente de sus etiquetas. También debe especificar el ID de recurso o nombre de recurso de Amazon (ARN), dependiendo de la operación que use.

Información adicional

Para obtener más información sobre el etiquetado, consulte los siguientes recursos.

- AWS Los principios de etiquetado en la referencia AWS general
- AWS Estrategias de etiquetado (PDF descargable)

- AWS El control de acceso en la guía del usuario AWS de IAM
- AWS Políticas de etiquetado en la guía del usuario AWS Organizations

Adición de etiquetas a recursos de Amazon Personalize

Puede añadir, mostrar, actualizar y eliminar claves y valores de etiquetas de los recursos de Amazon Personalize con la consola Amazon Personalize, AWS Command Line Interface (AWS CLI) o AWS SDKs. Los siguientes ejemplos muestran cómo agregar una etiqueta al grupo de conjuntos de datos de Amazon Personalize. Puede agregar etiquetas a otros recursos de Amazon Personalize del mismo modo.

Temas

- Adición de etiquetas (consola)
- Adición de etiquetas (AWS CLI)
- Adición de etiquetas (AWS SDKs)

Adición de etiquetas (consola)

Al crear un recurso en Amazon Personalize, puede añadir etiquetas opcionales con la consola de Amazon Personalize. El siguiente ejemplo añade una etiqueta a un grupo de conjuntos de datos.

Para agregar etiquetas a un grupo de conjunto de datos nuevo

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. Elija Crear grupo de conjuntos de datos.
3. En Nombre, ingrese un nombre.
4. En Dominio, elija un dominio.
5. Amplíe la sección Etiquetas, elija Añadir nueva etiqueta.
6. Escriba un valor adecuado para Clave y Valor.

   Por ejemplo, **`Environment`** y **`Test`**, respectivamente.
7. Para añadir más etiquetas, elija Añadir nueva etiqueta.

   Puede agregar hasta 50 etiquetas a un recurso.

8. Seleccione Siguiente para seguir creando el recurso.

Añadir etiquetas a un recurso existente es similar: elija su recurso y utilice los campos Etiquetas para añadir las etiquetas.

Adición de etiquetas (AWS CLI)

Puede usar AWS Command Line Interface (AWS CLI) para agregar etiquetas al crear un recurso o agregar etiquetas a un recurso existente.

Temas

- Adición de etiquetas al crear un recurso
- Adición de etiquetas a un recurso existente

Adición de etiquetas al crear un recurso

Para crear un nuevo recurso y añadirle una etiqueta con el AWS CLI, utilice el `create` comando correspondiente al recurso e incluya el `tags` parámetro y los valores. Por ejemplo, el siguiente comando crea un nuevo grupo de conjuntos de datos de dominio con el nombre `myDatasetGroup` del dominio ECOMMERCE y agrega las siguientes etiquetas: una clave de etiqueta `Environment` con un valor de etiqueta `Test` y una clave de etiqueta `Owner` y un valor `xyzCorp`.

```
aws personalize create-dataset-group \
--name myDatasetGroup \
--domain ECOMMERCE \
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

Para obtener información sobre los comandos que puede utilizar para crear un recurso de Amazon Personalize, consulte la referencia de AWS CLI comandos de Amazon Personalize.

Adición de etiquetas a un recurso existente

Para agregar una etiqueta a un recurso existente, utilice el comando `tag-resource`. Especifique el ARN del recurso y proporcione la clave y el valor de la etiqueta en el parámetro `tags`.

```
aws personalize tag-resource \
--resource-arn resource ARN \
```

```
--tags tagKey=key,tagValue=value
```

Adición de etiquetas (AWS SDKs)

Puede utilizarla AWS SDKs para añadir etiquetas al crear un recurso o para añadir etiquetas a un recurso existente.

Temas

- Adición de etiquetas al crear un recurso
- Adición de etiquetas a un recurso existente

Adición de etiquetas al crear un recurso

Para crear un recurso nuevo y añadirle una etiqueta con el AWS SDKs, utilice el `create` método adecuado. Use el parámetro `tags` para especificar los pares clave-valor para cada una de sus etiquetas. Por ejemplo, el siguiente código crea un nuevo grupo de conjuntos de datos de dominio con el nombre `myDatasetGroup` del dominio ECOMMERCE y agrega las siguientes etiquetas: una clave de etiqueta `Environment` con un valor de etiqueta `Test` y una clave de etiqueta `Owner` y un valor `xyzCorp`.

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 "";
}
```

Adición de etiquetas a un recurso existente

El siguiente código muestra cómo agregar una etiqueta a un recurso de Amazon Personalize existente. Especifique el nombre de recurso de Amazon (ARN) del recurso al que desea agregar etiquetas y especifique pares de clave-valor para cada una de sus etiquetas.

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 "";
    }
```

Eliminación de etiquetas de los recursos de Amazon Personalize

Puede eliminar etiquetas de los recursos de Amazon Personalize con la consola Amazon Personalize o la operación de la UntagResource API con AWS Command Line Interface (AWS CLI) o AWS SDKs. En los siguientes ejemplos se muestra cómo eliminar una etiqueta de un grupo de conjuntos de datos de Amazon Personalize. Puede eliminar etiquetas de otros recursos de Amazon Personalize del mismo modo.

Temas

- Eliminación de etiquetas (consola)
- Eliminación de etiquetas (AWS CLI)
- Eliminación de etiquetas (AWS SDKs)

Eliminación de etiquetas (consola)

Después de agregar etiquetas a un recurso en Amazon Personalize, puede eliminarlas con la consola de Amazon Personalize. El siguiente ejemplo elimina una etiqueta de un grupo de conjuntos de datos

Eliminación de etiquetas de un grupo de conjuntos de datos

1. Abre la consola Amazon Personalize en https://console.aws.amazon.com/personalize/casa e inicia sesión en tu cuenta.
2. Elija su grupo de conjuntos de datos.
3. En la parte inferior de la página, elija la pestaña Etiquetas y Administrar etiquetas.
4. Para cada etiqueta que desee eliminar, elija Eliminar.
5. Seleccione Guardar para eliminar las etiquetas.

Eliminación de etiquetas (AWS CLI)

Para eliminar etiquetas de un recurso existente con el AWS CLI, utilice el siguiente `untag-resource` comando. Para `resource-arn`, especifique el nombre de recurso de Amazon (ARN) del recurso. En `tag-keys`, especifique las claves de las etiquetas que desea eliminar.

```
aws personalize untag-resource \
--resource-arn resource ARN \
--tag-keys key1 key2
```

Eliminación de etiquetas (AWS SDKs)

Para eliminar etiquetas de un recurso Amazon Personalize existente con la AWS SDKs, utilice la operación de UntagResource API. El código siguiente muestra cómo eliminar varias etiquetas de un grupo de conjuntos de datos con el SDK para Python (Boto3). Para `resourceArn`, especifique el nombre de recurso de Amazon (ARN) del recurso. En `tagKeys`, especifique las claves de las etiquetas que desea eliminar.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.untag_resource(
    resourceArn="Resource ARN",
    tagKeys=["tag1Key", "tag2Key"]
)
```

Uso de etiquetas en políticas de IAM

Después de empezar a implementar etiquetas, puede aplicar permisos de nivel de recursos basados en etiquetas a las políticas de AWS Identity and Access Management (IAM) y las operaciones de API. Esto incluye operaciones que admiten agregar etiquetas a los recursos cuando se crean recursos. Al usar las etiquetas de esta manera, puede implementar un control pormenorizado sobre qué grupos y usuarios de su AWS cuenta tienen permiso para crear y etiquetar recursos, y qué grupos y usuarios tienen permiso para crear, actualizar y eliminar etiquetas de manera más general.

Por ejemplo, puede crear una política que permita a un usuario tener acceso completo a todos los recursos de Amazon Personalize en los que su nombre sea un valor en la etiqueta `Owner` del recurso:

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "personalize:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

El siguiente ejemplo muestra cómo crear una política para permitir crear y eliminar un conjunto de datos. Estas operaciones solo se permiten si el nombre de dicho usuario es `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 define los permisos de nivel de recurso basados en etiquetas, estos entrarán en vigor inmediatamente. Esto significa que sus recursos están más seguros en cuanto se crean y que puede empezar a aplicar el uso de etiquetas de nuevos recursos rápidamente. También puede usar permisos de nivel de recurso para controlar las claves y valores de etiqueta que se pueden asociar a recursos nuevos y existentes. Para obtener más información, consulte Control del acceso mediante etiquetas en la AWS Guía del usuario de IAM de .

Preguntas frecuentes sobre Amazon Personalize

Las siguientes son respuestas a preguntas frecuentes relacionadas con la importación de datos, el entrenamiento, la implementación de modelos, las recomendaciones y los filtros en Amazon Personalize.

Para obtener más preguntas y respuestas, consulte la hoja de referencia de Amazon Personalize en el repositorio de ejemplos de Amazon Personalize.

Temas

- Importación y administración de datos
- Creación de una solución y una versión de la solución
- Implementación de modelos (campañas personalizadas)
- Recomendaciones
- Filtrado de recomendaciones

Importación y administración de datos

¿En qué formato deben estar mis datos masivos?

Los datos masivos deben estar en formato de valores separados por comas (CSV). La primera fila del archivo CSV debe contener los encabezados de las columnas. Los encabezados de columna del archivo CSV deben corresponder al esquema pare crear el conjunto de datos. Si sus datos incluyen caracteres codificados que no son ASCII, su archivo CSV debe estar codificado en formato UTF-8. No ponga los encabezados entre comillas (“). Los datos `TIMESTAMP` y `CREATION_TIMESTAMP` deben estar en formato de tiempo de UNIX. Para obtener más información acerca de los datos de fecha y hora, consulte Datos de marca temporal. Para obtener más información acerca de los esquemas, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Para ver las directrices completas de formato de datos, consulte Preparación de los datos de entrenamiento para Amazon Personalize. Si no está seguro de cómo formatear los datos, puede usar Amazon SageMaker AI Data Wrangler (Data Wrangler) para prepararlos. Para obtener más información, consulte Preparación e importación de datos masivos con Amazon SageMaker AI Data Wrangler.

¿Cuántos datos de entrenamiento necesito?

Para todos los casos de uso (grupos de conjuntos de datos de dominio) y recetas personalizadas, sus datos de interacciones deben tener lo siguiente:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Al menos 25 usuarios únicos IDs con al menos dos interacciones entre elementos para cada uno.

Para obtener recomendaciones de calidad, le sugerimos que tenga al menos 50 000 interacciones de elementos de al menos 1000 usuarios, con dos o más interacciones de elementos cada uno.

Puede empezar con un conjunto de datos de interacciones de elementos vacío y, cuando haya registrado suficientes datos, crear su recomendador (grupo de conjuntos de datos de dominio) o una versión de solución personalizada utilizando solo los nuevos eventos registrados. Algunas recetas y casos de uso pueden tener requisitos de datos adicionales. Para obtener información sobre los requisitos de casos de uso, consulte Elección de un caso de uso. Para obtener más información sobre los requisitos de recetas, consulte Elección de una receta.

¿Cómo actualizo los atributos de un elemento o usuario?

Use la consola de Amazon Personalize o las operaciones PutItems o PutUsers para importar un elemento o un usuario con el mismo identificador de elemento pero con los atributos modificados.

¿Cómo se elimina un elemento o un usuario?

Amazon Personalize no admite la eliminación de un elemento o usuario específico. Para asegurarse de que un elemento o un usuario no aparezca en las recomendaciones, use un filtro para excluir los elementos. Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

¿Cómo se elimina un esquema?

Puede eliminar un esquema solo con la operación DeleteSchema. No puede utilizar la consola de Amazon Personalize para eliminar un esquema.

Creación de una solución y una versión de la solución

¿Qué receta debo utilizar?

La receta de Amazon Personalize que utilice depende de su caso de uso. Para obtener información sobre cómo hacer coincidir los casos de uso con las recetas, consulte Elección de una receta. La hoja de referencia de Amazon Personalize también incluye información sobre casos de uso y recetas.

¿Con qué frecuencia debo realizar el entrenamiento?

Recomendamos utilizar un entrenamiento automático con una frecuencia de entrenamiento semanal como mínimo. El entrenamiento automático le facilita mantener la relevancia de la recomendación. La frecuencia de entrenamiento depende de los requisitos empresariales, de la receta que utilice y de la frecuencia con que importe los datos. Para obtener más información, consulte Configuración del entrenamiento automático. Para obtener información sobre cómo mantener la relevancia, consulte Mantenimiento de la relevancia de las recomendaciones.

¿Debo usar AutoML?

No, en su lugar le recomendamos que asocie su caso de uso con diferentes recetas de Amazon Personalize y que elija una receta. Para obtener información sobre cómo hacer coincidir los casos de uso con las recetas, consulte Elección de una receta.

Implementación de modelos (campañas personalizadas)

¿Qué debo establecer para minProvisionedTPS de mi campaña?

Un valor elevado de `minProvisionedTPS` aumentará los costos. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realiza un seguimiento de tu consumo mediante CloudWatch las métricas de Amazon y `minProvisionedTPS` auméntalas según sea necesario.

¿Cómo puedo controlar el coste de mis campañas?

El proyecto Amazon Personalize Monitor proporciona un CloudWatch panel de control, métricas personalizadas, alarmas de uso y funciones de optimización de costes para las campañas de Amazon Personalize. Consulte Amazon Personalize Monitor en el repositorio de muestras de Amazon Personalize.

¿Cómo configuro el rendimiento máximo de transacciones para una campaña?

Solo puede establecer el rendimiento mínimo de una campaña. Cuando crea una campaña de Amazon Personalize, especifica una capacidad de transacción dedicada para crear

recomendaciones en tiempo real para los usuarios de su aplicación. Si su TPS aumenta más allá de `minProvisionedTPS`, Amazon Personalize escala automáticamente la capacidad aprovisionada hacia arriba y hacia abajo, pero nunca por debajo de `minProvisionedTPS`. Para obtener más información, consulte Transacciones mínimas aprovisionadas por segundo y escalado automático.

Recomendaciones

¿Cómo puedo saber si mi modelo de Amazon Personalize está generando recomendaciones de calidad?

Evalúe el rendimiento de la versión de su solución con métricas en línea y fuera de línea (consulteEvaluación de una versión de solución de Amazon Personalize con métricas) y realizando pruebas en línea (como A/B testing). For more information about A/B las pruebas, consulte)Medir el impacto de las recomendaciones mediante A/B pruebas.

¿Cómo elimino mi trabajo de inferencia por lotes y por qué su estado es “activo”?

No puede eliminar los trabajos de inferencia por lotes. Cuando el estado de un trabajo de inferencia por lotes está activo, el trabajo está completo. Puede acceder a sus recomendaciones en el bucket o la carpeta de salida de Amazon S3. No incurrirá en costes adicionales por el trabajo de inferencia por lotes una vez finalizado el trabajo. Sin embargo, puede incurrir en cargos adicionales por parte de otros servicios, como Amazon S3, por el almacenamiento de datos de entrada y salida.

¿Por qué mi campaña respaldada por SIMS recomienda elementos que no son similares en función de los metadatos?

SIMS utiliza su conjunto de datos de interacciones de elementos para determinar la similitud, no los metadatos de los elementos, como el color o el precio. SIMS identifica la presencia simultánea del elemento en los historiales de los usuarios de su conjunto de datos de interacciones para recomendar elementos similares. Para obtener más información, consulte Receta SIMS.

¿Puedo obtener más de 500 elementos de una sola operación de GetRecommendations API?

500 es el número máximo de elementos que puede recuperar en un solo GetRecommendations. Este valor no se puede aumentar.

Filtrado de recomendaciones

¿Por qué mis recomendaciones no se filtran como se esperaba?

Esto se puede producir por diversas razones.

- Es posible que haya un problema con el formato o la sintaxis de la expresión de filtro. Para ver ejemplos de expresiones de filtro con el formato correcto, consulte Ejemplos de expresiones de filtro.
- Amazon Personalize considera hasta 100 de las interacciones más recientes por usuario y por tipo de evento. Esta es una cuota ajustable. Puede solicitar un aumento de cuota mediante la consola de Service Quotas. Si no importa las interacciones de elemento de un usuario durante tres meses, los filtros dejarán de tener en cuenta los datos históricos del usuario. Para tener en cuenta estos datos, debe volver a importar todo el historial de eventos del usuario.

Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

¿Cómo puedo eliminar los elementos ya comprados de las recomendaciones?

Para los grupos de conjuntos de datos de dominios de ECOMMERCE, si crea un recomendador con el caso de uso Recomendaciones o Los clientes que vieron X también vieron, Amazon Personalize filtra automáticamente los elementos que el usuario ha comprado en función del userId que especifique y de los eventos `Purchase`.

Para otros casos de uso de grupos de conjuntos de datos de dominios o recursos personalizados, use un filtro para eliminar los elementos comprados. Agregue un atributo de tipo de evento `Purchased` a sus datos, registre los eventos Purchase con la operación `PutItems` y cree un filtro que elimine los elementos comprados de las recomendaciones. Por ejemplo:

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("purchased")
```

Para obtener más información, consulte Recomendaciones de filtrado y segmentos de usuarios.

Mensajes de error comunes en Amazon Personalize

Las siguientes secciones enumeran y explican algunos de los mensajes que pueden surgir al utilizar Amazon Personalize.

Temas

- Importación y administración de datos
- Creación de una solución y una versión de la solución (recursos personalizados)
- Implementación de modelos (campañas personalizadas)
- Recomendadores (grupos de conjuntos de datos de dominios)
- Recomendaciones
- Filtrado de recomendaciones

Importación y administración de datos

Mensaje de error: Ubicación de datos no válida.

Asegúrese de utilizar la sintaxis correcta para la ubicación del bucket de Amazon S3. Para los trabajos de importación de conjuntos de datos, utilice la siguiente sintaxis para la ubicación de los datos en Amazon S3:

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

Si sus archivos CSV se encuentran en una carpeta y desea subir varios archivos con un trabajo de importación de conjuntos de datos, use esta sintaxis sin el nombre de archivo CSV.

Mensaje de error: se produjo un error (LimitExceededException) al llamar a la CreateDatasetImportJob operación: más de 5 recursos con el estado PENDING o IN_PROGRESS.

Puede tener un total de 5 trabajos de importación de conjuntos de datos pendientes o en curso por región. Esta cuota no se puede ajustar. Para obtener una lista completa de cuotas para Amazon Personalize, consulte Puntos de conexión y cuotas de Amazon Personalize.

Mensaje de error: No se pudo crear un trabajo de importación de datos para el conjunto de datos <dataset type>... Privilegios insuficientes para acceder a los datos en Amazon S3.

Proporcione a Amazon Personalize acceso a sus recursos de Amazon S3 adjuntando políticas de acceso al bucket de Amazon S3 y al rol de servicio de Amazon Personalize. Consulte Concesión de acceso a Amazon Personalize para los recursos de Amazon S3.

Si utilizas AWS Key Management Service (AWS KMS) para el cifrado, debes conceder permiso a Amazon Personalize y a tu rol de servicio Amazon Personalize IAM para usar tu clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Mensaje de error: No se pudo crear un trabajo de importación de datos para el conjunto de datos <dataset type>… Al CSV de entrada le faltan las columnas siguientes: [COLUMN_NAME, COLUMN_NAME].

Los datos que importe a Amazon Personalize, incluidos los nombres de los atributos y los tipos de datos, deben coincidir con el esquema del conjunto de datos de destino. Para obtener más información, consulte Creación de archivos JSON de esquema para los esquemas de Amazon Personalize.

Mensaje de error: La longitud no puede superar los <character limit> caracteres para <COLLUMN_NAME>. Si ningún valor supera el límite de caracteres, asegúrese de que sus datos sigan las pautas de formato que se indican en el https://docs.aws.amazon.com/personalize/ latest/dg/ data archivo -prep-formatting.html.

Asegúrese de que todos los valores de esta columna no superen el límite de caracteres. Si ningún valor supera el límite de caracteres, compruebe lo siguiente en los campos textuales anteriores:

- Asegúrese de que todos los datos textuales estén entre comillas dobles. Utilice el carácter \ para aplicar escape en cualquier comilla doble o carácter \ en los datos.
- Comprueba que todos los registros del archivo CSV estén en una sola línea.

Creación de una solución y una versión de la solución (recursos personalizados)

Mensaje de error: No se pudo crear. El conjunto de datos tiene menos de 25 usuarios con al menos 2 interacciones cada uno.

Debe importar más datos antes de poder entrenar el modelo. Los requisitos de datos mínimos para entrenar un modelo son:

- Como mínimo, 1000 registros de interacciones de elementos de los usuarios que interactúan con los elementos de su catálogo. Estas interacciones pueden provenir de importaciones masivas, de eventos transmitidos o ambos.
- Al menos 25 usuarios únicos IDs con al menos dos interacciones entre elementos para cada uno.

Para obtener recomendaciones en tiempo real, importe más datos con un trabajo de importación de conjuntos de datos o registre más eventos de interacción para sus usuarios con un rastreador de eventos y la operación PutEvents. Para obtener más información sobre el registro de eventos en tiempo real, consulte Registro de los eventos en tiempo real para influir en las recomendaciones.

Para obtener recomendaciones por lotes, importe sus datos con un trabajo de importación de conjuntos de datos cuando tenga más datos. Para obtener más información sobre cómo importar datos masivos, consulte Importación de datos de entrenamiento directamente a conjuntos de datos de Amazon Personalize.

Implementación de modelos (campañas personalizadas)

Error: No se puede crear una campaña. Más de 5 recursos en estado ACTIVE. Elimine algunos e inténtelo de nuevo.

Puede tener un total de 5 campañas de Amazon Personalize activas por grupo de conjuntos de datos. La cuota es ajustable y puede solicitar un aumento de cuota mediante la consola de Service Quotas. Para obtener una lista completa de límites y cuotas para Amazon Personalize, consulte Puntos de conexión y cuotas de Amazon Personalize.

Recomendadores (grupos de conjuntos de datos de dominios)

Error: El conjunto de datos tiene menos de 1000 interacciones después de filtrar por tipo de evento:<event type>

Los diferentes casos de uso requieren diferentes tipos de eventos. Sus datos deben tener un mínimo de 1000 eventos del tipo requerido para su caso de uso. Para obtener más información, consulte Elección de un caso de uso

Recomendaciones

Mensaje de error del trabajo de inferencia por lotes: Ruta de entrada de S3 no válida o Ruta de salida de S3 no válida

Asegúrese de utilizar la sintaxis correcta para la ubicación de entrada o salida de Amazon S3. Asegúrese también de que la ubicación de salida sea diferente de la de los datos de entrada. Debe ser una carpeta del mismo bucket de Amazon S3 o de un bucket diferente.

Utilice la siguiente sintaxis para la ubicación del archivo de entrada en Amazon S3: **`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>`**

Utilice la siguiente sintaxis para la carpeta de salida de Amazon S3: **`s3://amzn-s3-demo-bucket/<output folder name>/`**

Filtrado de recomendaciones

Mensaje de error: No se pudo crear el filtro. Símbolo de entrada no válido: $parameterName. No se permiten marcadores de posición con el operador NOT_IN.

No puede utilizar parámetros de marcador de posición en una expresión de filtro que utilice el operador NOT_IN. En su lugar, utilice el operador IN y utilice la acción opuesta: utilice Incluir en lugar de Excluir (o al revés).

Por ejemplo, si quiere usar `INCLUDE ItemID WHERE Items.GENRE NOT IN ($GENRE)`, puede usar `EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE)` y obtener los mismos resultados.

Para obtener más información acerca de los filtros, consulte Filtrar elementos de expresión.

Mensaje de error: No se pudo crear el filtro. Expresión no válida... al filtrar campos de tipo booleano

No puede crear expresiones de filtro que filtren utilizando valores de tipo booleano en su esquema. Para filtrar en función de valores booleanos, utilice un esquema con un campo de tipo `String` y utilice los valores `True` y `False` en sus datos. También puede usar el tipo `int` o `long` y los valores `0` y `1`.

Para obtener más información acerca de los filtros, consulte Filtrar elementos de expresión.

Especificar recursos con AWS CloudFormation

Amazon Personalize está integrado con AWS CloudFormation un servicio que le ayuda a modelar y configurar sus AWS recursos para que pueda dedicar menos tiempo a crear y administrar sus recursos e infraestructura. Crea una plantilla que describe todos los AWS recursos que puede especificar (como los grupos de conjuntos de datos de Amazon Personalize). CloudFormation a continuación, aprovisiona y configura esos recursos por usted.

Cuando la utilices CloudFormation, podrás reutilizar la plantilla para configurar tus recursos de Amazon Personalize de forma coherente y repetida. Describa sus recursos una vez y, a continuación, aprovisione los mismos recursos una y otra vez en varias Cuentas de AWS regiones.

Temas

- Amazon Personalize y CloudFormation plantillas
- CloudFormation Plantillas de ejemplo para los recursos de Amazon Personalize
- Obtenga más información sobre CloudFormation

Amazon Personalize y CloudFormation plantillas

Para aprovisionar y configurar los recursos de Amazon Personalize y los servicios relacionados, debe comprender las plantillas de CloudFormation. Las plantillas son archivos de texto con formato JSON o YAML. Estas plantillas describen los recursos que desea aprovisionar en sus CloudFormation pilas. Si no estás familiarizado con JSON o YAML, puedes usar CloudFormation Designer para ayudarte a empezar con CloudFormation las plantillas. Para obtener más información, consulte ¿Qué es Designer de CloudFormation ? en la Guía del usuario de AWS CloudFormation .

Amazon Personalize permite especificar conjuntos de datos, grupos de conjuntos de datos, trabajos de importación de conjuntos de datos, esquemas y soluciones en CloudFormation. Para obtener más información, consulte la Referencia de tipos de recursos de Amazon Personalize en la Guía del usuario de AWS CloudFormation .

CloudFormation Plantillas de ejemplo para los recursos de Amazon Personalize

Los siguientes ejemplos CloudFormation de plantillas muestran cómo especificar distintos recursos de Amazon Personalize.

Temas

- 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'
```

Obtenga más información sobre CloudFormation

Para obtener más información CloudFormation, consulte los siguientes recursos:

- AWS CloudFormation
- AWS CloudFormation guía del usuario
- CloudFormation Referencia de la API
- Guía del usuario de la interfaz de la línea de comandos de AWS CloudFormation

Ejemplos de código de Amazon Personalize usando SDK de AWS

En los siguientes ejemplos de código, se muestra cómo utilizar Amazon Personalize con un kit de desarrollo de software (SDK) de AWS.

Para obtener una lista completa de las guías para desarrolladores de AWS SDK y ejemplos de código, consulte Uso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos de código para Amazon Personalize usando AWS SDKs
  - Ejemplos básicos de Amazon Personalize usando AWS SDKs
    - Acciones para Amazon Personalize mediante AWS SDKs
      - CreateBatchInferenceJobÚselo con un AWS SDK
      - CreateBatchSegmentJobÚselo con un AWS SDK
      - CreateCampaignÚselo con un AWS SDK
      - CreateDatasetÚselo con un AWS SDK
      - CreateDatasetExportJobÚselo con un AWS SDK
      - CreateDatasetGroupÚselo con un AWS SDK
      - CreateDatasetImportJobÚselo con un AWS SDK
      - CreateEventTrackerÚselo con un AWS SDK
      - CreateFilterÚselo con un AWS SDK
      - CreateRecommenderÚselo con un AWS SDK
      - CreateSchemaÚselo con un AWS SDK
      - CreateSolutionÚselo con un AWS SDK
      - CreateSolutionVersionÚselo con un AWS SDK
      - DeleteCampaignÚselo con un AWS SDK
      - DeleteEventTrackerÚselo con un AWS SDK
      - DeleteSolutionÚselo con un AWS SDK
      - DescribeCampaignÚselo con un AWS SDK
      - DescribeRecipeÚselo con un AWS SDK

- DescribeSolutionÚselo con un AWS SDK
- ListCampaignsÚselo con un AWS SDK
- ListDatasetGroupsÚselo con un AWS SDK
- ListRecipesÚselo con un AWS SDK
- ListSolutionsÚselo con un AWS SDK
- UpdateCampaignÚselo con un AWS SDK

- Ejemplos de código para Amazon Personalize Events mediante AWS SDKs
  - Ejemplos básicos de Amazon Personalize Events mediante AWS SDKs
    - Acciones para Amazon Personalize Events mediante AWS SDKs
      - PutEventsUtilízalo con un AWS SDK
      - PutItemsÚselo con un AWS SDK
      - PutUsersÚselo con un AWS SDK
- Ejemplos de código para Amazon Personalize Runtime mediante AWS SDKs
  - Ejemplos básicos de Amazon Personalize Runtime mediante AWS SDKs
    - Acciones para Amazon Personalize Runtime mediante AWS SDKs
      - GetPersonalizedRankingÚselo con un AWS SDK
      - GetRecommendationsÚselo con un AWS SDK

Ejemplos de código para Amazon Personalize usando AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar Amazon Personalize con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Amazon Personalize

- Ejemplos básicos de Amazon Personalize usando AWS SDKs
  - Acciones para Amazon Personalize mediante AWS SDKs

- CreateBatchInferenceJobÚselo con un AWS SDK
- CreateBatchSegmentJobÚselo con un AWS SDK
- CreateCampaignÚselo con un AWS SDK
- CreateDatasetÚselo con un AWS SDK
- CreateDatasetExportJobÚselo con un AWS SDK
- CreateDatasetGroupÚselo con un AWS SDK
- CreateDatasetImportJobÚselo con un AWS SDK
- CreateEventTrackerÚselo con un AWS SDK
- CreateFilterÚselo con un AWS SDK
- CreateRecommenderÚselo con un AWS SDK
- CreateSchemaÚselo con un AWS SDK
- CreateSolutionÚselo con un AWS SDK
- CreateSolutionVersionÚselo con un AWS SDK
- DeleteCampaignÚselo con un AWS SDK
- DeleteEventTrackerÚselo con un AWS SDK
- DeleteSolutionÚselo con un AWS SDK
- DescribeCampaignÚselo con un AWS SDK
- DescribeRecipeÚselo con un AWS SDK
- DescribeSolutionÚselo con un AWS SDK
- ListCampaignsÚselo con un AWS SDK
- ListDatasetGroupsÚselo con un AWS SDK
- ListRecipesÚselo con un AWS SDK
- ListSolutionsÚselo con un AWS SDK
- UpdateCampaignÚselo con un AWS SDK

Ejemplos básicos de Amazon Personalize usando AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Personalize with AWS SDKs.

Ejemplos

- Acciones para Amazon Personalize mediante AWS SDKs

- CreateBatchInferenceJobÚselo con un AWS SDK
- CreateBatchSegmentJobÚselo con un AWS SDK
- CreateCampaignÚselo con un AWS SDK
- CreateDatasetÚselo con un AWS SDK
- CreateDatasetExportJobÚselo con un AWS SDK
- CreateDatasetGroupÚselo con un AWS SDK
- CreateDatasetImportJobÚselo con un AWS SDK
- CreateEventTrackerÚselo con un AWS SDK
- CreateFilterÚselo con un AWS SDK
- CreateRecommenderÚselo con un AWS SDK
- CreateSchemaÚselo con un AWS SDK
- CreateSolutionÚselo con un AWS SDK
- CreateSolutionVersionÚselo con un AWS SDK
- DeleteCampaignÚselo con un AWS SDK
- DeleteEventTrackerÚselo con un AWS SDK
- DeleteSolutionÚselo con un AWS SDK
- DescribeCampaignÚselo con un AWS SDK
- DescribeRecipeÚselo con un AWS SDK
- DescribeSolutionÚselo con un AWS SDK
- ListCampaignsÚselo con un AWS SDK
- ListDatasetGroupsÚselo con un AWS SDK
- ListRecipesÚselo con un AWS SDK
- ListSolutionsÚselo con un AWS SDK
- UpdateCampaignÚselo con un AWS SDK

Acciones para Amazon Personalize mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Personalize con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de API de Amazon Personalize.

Ejemplos

- CreateBatchInferenceJobÚselo con un AWS SDK
- CreateBatchSegmentJobÚselo con un AWS SDK
- CreateCampaignÚselo con un AWS SDK
- CreateDatasetÚselo con un AWS SDK
- CreateDatasetExportJobÚselo con un AWS SDK
- CreateDatasetGroupÚselo con un AWS SDK
- CreateDatasetImportJobÚselo con un AWS SDK
- CreateEventTrackerÚselo con un AWS SDK
- CreateFilterÚselo con un AWS SDK
- CreateRecommenderÚselo con un AWS SDK
- CreateSchemaÚselo con un AWS SDK
- CreateSolutionÚselo con un AWS SDK
- CreateSolutionVersionÚselo con un AWS SDK
- DeleteCampaignÚselo con un AWS SDK
- DeleteEventTrackerÚselo con un AWS SDK
- DeleteSolutionÚselo con un AWS SDK
- DescribeCampaignÚselo con un AWS SDK
- DescribeRecipeÚselo con un AWS SDK
- DescribeSolutionÚselo con un AWS SDK
- ListCampaignsÚselo con un AWS SDK
- ListDatasetGroupsÚselo con un AWS SDK
- ListRecipesÚselo con un AWS SDK
- ListSolutionsÚselo con un AWS SDK
- UpdateCampaignÚselo con un AWS SDK

`CreateBatchInferenceJob`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateBatchInferenceJob`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
        }
```

- Para obtener más información sobre la API, consulta CreateBatchInferenceJobla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateBatchInferenceJobla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateBatchSegmentJob`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateBatchSegmentJob`.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateBatchSegmentJobla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateCampaign`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
   }
```

- Para obtener más información sobre la API, consulta CreateCampaignla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateCampaignla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateDataset`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateDataset`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
}
```

- Para obtener más información sobre la API, consulta CreateDatasetla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateDatasetla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateDatasetExportJob`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateDatasetExportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateDatasetExportJobla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateDatasetExportJobla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateDatasetGroup`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateDatasetGroup`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

Crear un grupo de conjuntos de datos de dominios.

```
    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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateDatasetGroupla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
```

Crear un grupo de conjuntos de datos de dominios.

```
// 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();
```

- Para obtener más información sobre la API, consulta CreateDatasetGroupla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateDatasetImportJob`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateDatasetImportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateDatasetImportJobla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateDatasetImportJobla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateEventTracker`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateEventTracker`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
```

- Para obtener más información sobre la API, consulta CreateEventTrackerla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateEventTrackerla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateFilter`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateFilter`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
}
```

- Para obtener más información sobre la API, consulta CreateFilterla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateFilterla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateRecommender`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateRecommender`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateRecommenderla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateRecommenderla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateSchema`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateSchema`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

Cree un esquema con un dominio.

```
    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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateSchemala Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

Cree un esquema con un dominio.

```
// 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();
```

- Para obtener más información sobre la API, consulta CreateSchemala Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateSolution`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateSolution`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateSolutionla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateSolutionla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateSolutionVersion`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateSolutionVersion`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta CreateSolutionVersionla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta CreateSolutionVersionla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteCampaign`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
}
```

- Para obtener más información sobre la API, consulta DeleteCampaignla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteEventTracker`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteEventTracker`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta DeleteEventTrackerla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteSolution`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteSolution`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta DeleteSolutionla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeCampaign`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta DescribeCampaignla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeRecipe`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeRecipe`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta DescribeRecipela Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeSolution`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeSolution`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta DescribeSolutionla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListCampaigns`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListCampaigns`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta ListCampaignsla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListDatasetGroups`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDatasetGroups`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener más información sobre la API, consulta ListDatasetGroupsla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListRecipes`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListRecipes`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener más información sobre la API, consulta ListRecipesla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListSolutions`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListSolutions`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener más información sobre la API, consulta ListSolutionsla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`UpdateCampaign`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
```

- Para obtener más información sobre la API, consulta UpdateCampaignla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para Amazon Personalize Events mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar Amazon Personalize Events con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Eventos de Amazon Personalize

- Ejemplos básicos de Amazon Personalize Events mediante AWS SDKs
  - Acciones para Amazon Personalize Events mediante AWS SDKs
    - PutEventsUtilízalo con un AWS SDK
    - PutItemsÚselo con un AWS SDK
    - PutUsersÚselo con un AWS SDK

Ejemplos básicos de Amazon Personalize Events mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Personalize Events con AWS SDKs.

Ejemplos

- Acciones para Amazon Personalize Events mediante AWS SDKs
  - PutEventsUtilízalo con un AWS SDK
  - PutItemsÚselo con un AWS SDK
  - PutUsersÚselo con un AWS SDK

Acciones para Amazon Personalize Events mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Personalize Events con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de API de Eventos de Amazon Personalize.

Ejemplos

- PutEventsUtilízalo con un AWS SDK
- PutItemsÚselo con un AWS SDK
- PutUsersÚselo con un AWS SDK

`PutEvents`Utilízalo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `PutEvents`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`PutItems`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `PutItems`.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta PutItemsla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`PutUsers`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `PutUsers`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener más información sobre la API, consulta PutUsersla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta PutUsersla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para Amazon Personalize Runtime mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar Amazon Personalize Runtime con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Versión ejecutable de Amazon Personalize

- Ejemplos básicos de Amazon Personalize Runtime mediante AWS SDKs
  - Acciones para Amazon Personalize Runtime mediante AWS SDKs
    - GetPersonalizedRankingÚselo con un AWS SDK
    - GetRecommendationsÚselo con un AWS SDK

Ejemplos básicos de Amazon Personalize Runtime mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Personalize Runtime con AWS SDKs.

Ejemplos

- Acciones para Amazon Personalize Runtime mediante AWS SDKs
  - GetPersonalizedRankingÚselo con un AWS SDK
  - GetRecommendationsÚselo con un AWS SDK

Acciones para Amazon Personalize Runtime mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Personalize Runtime con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API del tiempo de ejecución de Amazon Personalize.

Ejemplos

- GetPersonalizedRankingÚselo con un AWS SDK
- GetRecommendationsÚselo con un AWS SDK

`GetPersonalizedRanking`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `GetPersonalizedRanking`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
```

- Para obtener más información sobre la API, consulta GetPersonalizedRankingla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener más información sobre la API, consulta GetPersonalizedRankingla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`GetRecommendations`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `GetRecommendations`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtener una lista de artículos recomendados.

```
   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);
        }
    }
```

Obtenga una lista de recomendaciones de un recomendador creado en un grupo de conjuntos de datos de dominio.

```
    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);
        }
    }
```

Use un filtro cuando solicite recomendaciones.

```
    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);
        }
    }
```

- Para obtener más información sobre la API, consulta GetRecommendationsla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

Obtener recomendaciones con un filtro (grupo de conjuntos de datos personalizados).

```
// 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();
```

Obtenga recomendaciones filtradas de un recomendador creado en un grupo de conjuntos de datos de dominio.

```
// 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();
```

- Para obtener más información sobre la API, consulta GetRecommendationsla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Personalize con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en Amazon Personalize

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El https://aws.amazon.com/compliance/shared-responsibility-model/ describe estos conceptos como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Amazon Personalize utiliza el cifrado de datos para proteger sus datos. Para obtener más información, consulteCifrado de datos en Amazon Personalize. Third-partylos auditores prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los programas de AWS cumplimiento de los programas de . Para obtener información sobre los programas de conformidad que se aplican a Amazon Personalize, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le permite comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Personalize. En los siguientes temas, se mostrará cómo configurar Amazon Personalize para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de Amazon Personalize.

Temas

- Protección de datos en Amazon Personalize
- Administración de identidades y accesos para Amazon Personalize
- Supervisión de Amazon Personalize con Amazon CloudWatch
- Registro de llamadas a la API Amazon Personalize con AWS CloudTrail
- Validación de conformidad para Amazon Personalize
- Resiliencia de Amazon Personalize

- Seguridad de la infraestructura en Amazon Personalize
- Amazon Personalize e interconecta los puntos de enlace de VPC (AWS PrivateLink)

Protección de datos en Amazon Personalize

El modelo de se aplica a la protección de datos en Amazon Personalize. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre privacidad de datos y los . Para obtener más información sobre la protección de datos en Europa, consulte el Centro del Reglamento General de Protección de Datos (RGPD).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contienen Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon Personalize u otro tipo de Servicios de AWS uso de la consola, la API o AWS los SDK. AWS CLI Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado de datos en Amazon Personalize

La siguiente información explica dónde Amazon Personalize utiliza el cifrado de datos para proteger los datos.

Cifrado en reposo

Todos los datos almacenados en Amazon Personalize siempre se cifran en reposo con las claves gestionadas por Amazon Personalize AWS Key Management Service (AWS KMS). Si proporciona su propia clave de AWS KMS durante la creación del recurso, Amazon Personalize utiliza la clave para cifrar los datos y almacenarlos. Por ejemplo, si proporciona un ARN de AWS KMS en la operación CreateDatasetGroup, Amazon Personalize utiliza la clave para cifrar y almacenar los datos que importe en cualquier conjunto de datos que cree en ese grupo de conjuntos de datos.

Debe conceder permiso a Amazon Personalize y a su rol de servicio de IAM de Amazon Personalize para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Para obtener información acerca del cifrado de datos de Amazon S3, consulte Protección de datos mediante cifrado en la Guía del usuario de Amazon Simple Storage Service. Para obtener información sobre cómo administrar su propia AWS KMS clave, consulte Administración de claves en la Guía para AWS Key Management Service desarrolladores.

Cifrado en tránsito

Amazon Personalize utiliza TLS con AWS certificados para cifrar los datos que se envían a otros AWS servicios. Cualquier comunicación con otros AWS servicios se realiza a través de HTTPS, y los puntos de conexión de Amazon Personalize solo admiten conexiones seguras a través de HTTPS.

Amazon Personalize copia los datos de tu cuenta y los procesa en un AWS sistema interno. Al procesar los datos, Amazon Personalize los cifra con una AWS KMS clave Amazon Personalize o con cualquier AWS KMS clave que usted proporcione.

Administración de claves

AWS gestiona todas las AWS KMS claves predeterminadas. Es su responsabilidad gestionar todas AWS KMS las claves que posea. Debe conceder permiso a Amazon Personalize y a su rol de servicio de IAM de Amazon Personalize para usar su clave. Para obtener más información, consulte Otorgar permiso a Amazon Personalize para usar tu AWS KMS clave.

Para obtener información sobre cómo administrar tu propia AWS KMS clave, consulta Administrar claves en la Guía para AWS Key Management Service desarrolladores.

Administración de identidades y accesos para Amazon Personalize

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon Personalize. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración del acceso con políticas
- Cómo funciona Amazon Personalize con IAM
- Cross-service prevención confusa de diputados
- Identity-based ejemplos de políticas para Amazon Personalize
- Solución de problemas de identidad y acceso de Amazon Personalize

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte Solución de problemas de identidad y acceso de Amazon Personalize).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte Cómo funciona Amazon Personalize con IAM).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte Identity-based ejemplos de políticas para Amazon Personalize).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Al crear una Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz, que tiene acceso completo a todos los Servicios de AWS recursos. Se recomiendaencarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un Rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Identity-based políticas

Identity-based las políticas son documentos de política de permisos de JSON que se adjuntan a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Identity-based las políticas pueden ser políticas integradas (integradas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Selección entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Resource-based políticas

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe especificar una entidad principal en una política basada en recursos.

Resource-based las políticas son políticas en línea que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations .

- Políticas de control de recursos (RCP): definen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte Políticas de control de recursos (RCP) en la Guía del usuario de AWS Organizations .
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon Personalize con IAM

Antes de utilizar IAM para administrar el acceso a Amazon Personalize, obtenga información sobre qué características de IAM se encuentran disponibles con Amazon Personalize.

Características de IAM que puede utilizar con Amazon Personalize

Característica de IAM	Compatibilidad de Amazon Personalize
Identity-based políticas	Sí
Resource-based políticas	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí

Característica de IAM	Compatibilidad de Amazon Personalize
Permisos de entidades principales	Sí
Roles de servicio	Sí
Service-linked roles	No

Para obtener una visión general de cómo funcionan Amazon Personalize y otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Identity-based políticas de Amazon Personalize

Compatibilidad con las políticas basadas en identidad: sí

Identity-based las políticas son documentos de política de permisos de JSON que puedes adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de la política de JSON de IAM en la Guía del usuario de IAM.

Identity-based ejemplos de políticas para Amazon Personalize

Para ver ejemplos de políticas basadas en identidades de Amazon Personalize, consulte Identity-based ejemplos de políticas para Amazon Personalize.

Resource-based políticas de Amazon Personalize

Admite políticas basadas en recursos: no

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las

políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política basada en recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones de políticas para Amazon Personalize

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon Personalize, consulte Acciones definidas por Amazon Personalize en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amazon Personalize utilizan el siguiente prefijo antes de la acción:

```
personalize
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "personalize:action1",
      "personalize:action2"
         ]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "personalize:Describe*"
```

Para ver ejemplos de políticas basadas en identidades de Amazon Personalize, consulte Identity-based ejemplos de políticas para Amazon Personalize.

Recursos de políticas para Amazon Personalize

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de tipos de recursos de Amazon Personalize y sus ARN, consulte Tipos de recurso definidos por Amazon Personalize en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon Personalize.

Para ver ejemplos de políticas basadas en identidades de Amazon Personalize, consulte Identity-based ejemplos de políticas para Amazon Personalize.

Claves de condición de políticas para Amazon Personalize

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para

ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de las claves de condición de Amazon Personalize, consulte Claves de condición para Amazon Personalize en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Personalize.

Para ver ejemplos de políticas basadas en identidades de Amazon Personalize, consulte Identity-based ejemplos de políticas para Amazon Personalize.

ACL en Amazon Personalize

Compatibilidad con ACL: no

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con Amazon Personalize

Admite ABAC (etiquetas en las políticas): sí

Attribute-based el control de acceso (ABAC) es una estrategia de autorización que define los permisos en función de unos atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del director coincida con la etiqueta del recurso.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Para obtener más información acerca del etiquetado de recursos de Amazon Personalize, consulte Etiquetado de recursos de Amazon Personalize.

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte Uso de etiquetas en políticas de IAM.

Uso de credenciales temporales con Amazon Personalize

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM y Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Cross-service permisos principales para Amazon Personalize

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos del operador principal que realiza la llamada Servicio de AWS, junto con los de solicitud, Servicio de AWS para realizar solicitudes a los servicios descendentes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Sesiones de acceso directo.

Roles de servicio para Amazon Personalize

Compatible con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Amazon Personalize. Edite los roles de servicio solo cuando Amazon Personalize proporcione orientación para hacerlo.

Service-linked roles para Amazon Personalize

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir la función de realizar una acción en su nombre. Service-linked las funciones aparecen en su nombre Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya una `Yes` en la columna de Service-linked funciones. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Cross-service prevención confusa de diputados

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. Cross-service la suplantación de identidad puede producirse cuando un servicio (el servicio de llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición global `aws:SourceArn` o `aws:SourceAccount` en las políticas de recursos para limitar los permisos que Amazon Personalize concede a otro servicio para el recurso.

Para evitar el problema del suplente confuso en los roles asumidos por Amazon Personalize, en la política de confianza del rol, establezca el valor de `aws:SourceArn` en `arn:aws:personalize:`*`region`*`:`*`accountNumber`*`:*`. El comodín (`*`) aplica la condición a todos los recursos de Amazon Personalize.

La siguiente política de relación de confianza concede a Amazon Personalize accesos a sus recursos y usa las claves de contexto de condiciones globales `aws:SourceArn` y `aws:SourceAccount` para evitar el problema del suplente confuso. Use esta política al crear un rol para Amazon Personalize (Creación de un rol de IAM para 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 ejemplos de políticas para Amazon Personalize

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon Personalize. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

A fin de obtener más información sobre las acciones y los tipos de recursos definidos por Amazon Personalize, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición de Amazon Personalize en la Referencia de autorizaciones de servicio.

Temas

- Prácticas recomendadas sobre las políticas
- AWS políticas administradas

- Uso de la consola de Amazon Personalize
- Cómo permitir a los usuarios consultar sus propios permisos
- Permiso del acceso total a los recursos de Amazon Personalize
- Permiso del acceso de solo lectura a recursos de Amazon Personalize

Prácticas recomendadas sobre las políticas

Identity-based las políticas determinan si alguien puede crear, acceder o eliminar los recursos de Amazon Personalize de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para

más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

AWS políticas administradas

AWS las políticas administradas son políticas creadas y administradas por. AWS Los siguientes son ejemplos de políticas AWS gestionadas que puede utilizar al trabajar con Amazon Personalize.

AmazonPersonalizeFullAccess Política

Puede utilizar la `AmazonPersonalizeFullAccess` política AWS gestionada para conceder a los usuarios los siguientes permisos:

- Acceder a todos los recursos de Amazon Personalize
- Publica y publica métricas en Amazon CloudWatch
- Enumerar, leer, escribir y eliminar todos los objetos de un bucket de Amazon S3 que contiene `Personalize` o `personalize` en el nombre del bucket
- Pasar un rol a Amazon Personalize

`AmazonPersonalizeFullAccess` proporciona más permisos de los necesarios. Recomendamos crear una nueva política de IAM que solo conceda los permisos necesarios (consulte Concesión de permiso a Amazon Personalize para que acceda a sus recursos).

CloudWatchFullAccess

Para dar permiso a tus usuarios para que supervisen Amazon Personalize con CloudWatch, adjunta la `CloudWatchFullAccess` política a tu función. Para obtener más información, consulte Supervisión de Amazon Personalize con Amazon CloudWatch.

La política de `CloudWatchFullAccess` es opcional y concede permiso para la siguientes acciones:

- Publica y enumera las métricas de Amazon Personalize en CloudWatch

- Ver las métricas y las estadísticas de las métricas
- Configurar alarmas basadas en métricas

Uso de la consola de Amazon Personalize

Para acceder a la consola de Amazon Personalize, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirte enumerar y ver detalles sobre los recursos de Amazon Personalize en tu Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o 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": "*"
        }
    ]
}
```

Permiso del acceso total a los recursos de Amazon Personalize

El siguiente ejemplo proporciona a un usuario de IAM de su AWS cuenta acceso completo a todos los recursos y acciones de Amazon Personalize.

JSON

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

Permiso del acceso de solo lectura a recursos de Amazon Personalize

En este ejemplo, concedes a un usuario de IAM de tu AWS cuenta acceso de solo lectura a tus recursos de Amazon Personalize, incluidos los conjuntos de datos, los grupos de conjuntos de datos, las soluciones y las campañas de Amazon Personalize.

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": "*"
        }
    ]
```

```
}
```

Solución de problemas de identidad y acceso de Amazon Personalize

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que es posible que surjan cuando se trabaja con Amazon Personalize e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon Personalize
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mi Cuenta de AWS para acceder a mis recursos de Amazon Personalize

No tengo autorización para realizar una acción en Amazon Personalize

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `personalize:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 personalize:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `personalize:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, sus políticas deben actualizarse para permitirle pasar un rol a Amazon Personalize.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Personalize. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi Cuenta de AWS para acceder a mis recursos de Amazon Personalize

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amazon Personalize admite estas características, consulte Cómo funciona Amazon Personalize con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Supervisión de Amazon Personalize con Amazon CloudWatch

Con Amazon CloudWatch, puedes obtener las métricas asociadas a Amazon Personalize. Puede configurar alarmas que le avisen cuando una o varias de estas métricas superen un umbral definido. Para ver las métricas, puedes usar Amazon CloudWatch AWS Command Line Interface, Amazon o la CloudWatch API.

Temas

- Uso de CloudWatch métricas para Amazon Personalize
- Acceso a métricas de Amazon Personalize
- Crear una alarma de
- Ejemplo de aplicación de monitorización de tecnología sin servidor de Amazon Personalize
- CloudWatch métricas de Amazon Personalize

Uso de CloudWatch métricas para Amazon Personalize

Para utilizar métricas, debe especificar la siguiente información:

- El nombre de la métrica.
- La dimensión de la métrica. Una dimensión es un par de nombre-valor que le ayuda a identificar una métrica de forma inequívoca.

Puede obtener datos de supervisión de Amazon Personalize mediante la Consola de administración de AWS AWS CLI, la o la CloudWatch API. También puede usar la CloudWatch API a través de uno de los AWS SDK o las herramientas de la CloudWatch API. La consola muestra una serie de gráficos basados en los datos sin procesar de la CloudWatch API. En función de sus necesidades, es posible que prefiera utilizar los gráficos que se muestran en la consola o que se recuperan de la API.

En la siguiente lista se indican algunos usos frecuentes de las métricas. Se trata de sugerencias que puede usar como punto de partida y no de una lista completa.

¿Cómo?	Métrica relevante
¿Cómo puedo monitorear el número de eventos que se han registrado?	Monitorice la métrica `PutEventsRequests` .
¿Cómo puedo supervisar los DatasetImportJob errores?	Utilice la métrica `DatasetImportJobError` .
¿Cómo puedo monitorizar la latencia de las llamadas `GetRecommendations` ?	Utilice la métrica `GetRecommendations Latency` .

Debes tener los CloudWatch permisos adecuados para monitorizar Amazon Personalize with CloudWatch. Para obtener más información, consulta Autenticación y control de acceso para Amazon CloudWatch.

Acceso a métricas de Amazon Personalize

Los siguientes ejemplos muestran cómo acceder a las métricas de Amazon Personalize mediante la CloudWatch consola AWS CLI, la y la CloudWatch API.

Para ver las métricas (consola)

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. Elija Métricas, elija la pestaña Todas las métricas y, a continuación, elija `AWS/Personalize`.
3. Elija la dimensión de la métrica.
4. Elija en la lista la métrica que desea usar y elija un periodo de tiempo para el gráfico.

Para ver las métricas de los eventos recibidos durante un periodo de tiempo (CLI)

- Abre AWS CLI e introduce el siguiente comando:

```
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
```

Este ejemplo muestra los eventos recibidos para el ARN del rastreador de eventos durante un periodo de tiempo. Para obtener más información, consulte get-metric-statistics.

Para acceder a las métricas (CloudWatch API)

- Llamar a `GetMetricStatistics`. Para obtener más información, consulta la referencia de la CloudWatch API de Amazon.

Crear una alarma de

Puede crear una CloudWatch alarma que envíe un mensaje de Amazon Simple Notification Service (Amazon SNS) cuando la alarma cambie de estado. Una alarma vigila una métrica individual durante un periodo de tiempo que usted especifica. La alarma realiza una o varias acciones según el valor de la métrica con respecto a un umbral determinado durante varios períodos de tiempo. La acción es una notificación que se envía a un tema de Amazon SNS o a una política de AWS Auto Scaling .

Las alarmas invocan acciones únicamente en caso de cambios de estado sostenidos. CloudWatch las alarmas no invocan acciones simplemente porque se encuentran en un estado determinado. El estado debe haber cambiado y debe mantenerse durante el número de periodos de tiempo especificado.

Para configurar una alarma (consola)

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarmas y, a continuación, elija Crear alarma. Esto lanza el Asistente de creación de alarmas.
3. Elija Seleccionar métrica.
4. En la pestaña Todas las métricas, elija `AWS/Personalize`.
5. Elige y EventTrackerArn, a continuación, elige PutEventsRequestslas métricas.
6. Elija la pestaña Métricas diagramadas.

7. En Estadística, elija Suma.
8. Elija Seleccionar métrica.
9. Rellene Nombre y Descripción. En Cuando, elija **>** y, a continuación, escriba un valor máximo de su elección.
10. Si CloudWatch quieres enviarte un correo electrónico cuando se alcance el estado de alarma, en Siempre que esta alarma:, selecciona State is ALARM. Para enviar alarmas a un tema de Amazon SNS existente, en Enviar notificación a:, elija un tema de SNS existente. Para configurar el nombre y las direcciones de correo electrónico de una nueva lista de suscripciones de correo electrónico, selecciona Nueva lista. CloudWatch guarda la lista y la muestra en el campo para que pueda utilizarla para configurar futuras alarmas.

    > **Note**
    >
    > Si utiliza Nueva lista para crear un nuevo tema de Amazon SNS, deben verificarse las direcciones de correo electrónico para que los destinatarios previstos puedan recibir las notificaciones. Amazon SNS envía solo mensajes de correo electrónico cuando la alarma entra en un estado de alarma. Si este cambio en el estado de la alarma se produce antes de que se verifiquen las direcciones de correo electrónico, los destinatarios no reciben una notificación.

11. Elija Crear alarma.

Para configurar una alarma (AWS CLI)

- Abra y AWS CLI, a continuación, introduzca el siguiente comando. Cambie el valor del parámetro `alarm-actions` de forma que haga referencia a un tema de Amazon SNS que haya creado anteriormente.

```
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
```

Este ejemplo muestra cómo crear una alarma cuando se producen más de 10 eventos para el ARN del rastreador de eventos en 5 minutos. Para obtener más información, consulte put-metric-alarm.

Para configurar una alarma (CloudWatch API)

- Llamar a `PutMetricAlarm`. Para obtener más información, consulta Amazon CloudWatch API Reference.

Ejemplo de aplicación de monitorización de tecnología sin servidor de Amazon Personalize

Para ver un ejemplo de aplicación que añade funciones de supervisión, alertas y optimización para Amazon Personalize, consulte Amazon Personalize monitor en el repositorio de Amazon Personalize samples.

CloudWatch métricas de Amazon Personalize

Esta sección contiene información sobre las CloudWatch métricas de Amazon disponibles para Amazon Personalize. Para obtener más información, consulte Supervisión de Amazon Personalize con Amazon CloudWatch.

En la siguiente tabla se enumeran las métricas de Amazon Personalize. Todas las métricas excepto GetRecommendations y GetPersonalizedRanking compatibles con estas estadísticas:`Average, Minimum, Maximum, Sum`. GetRecommendations y `Sum` solo GetPersonalizedRanking soporte.

Métrica	Description (Descripción)
`DatasetImportJobRequests`	El número de llamadas correctas a la API CreateDatasetImportJob. Dimensiones: `DatasetGroupArn, DatasetArn, DatasetImportJobArn`

Métrica	Description (Descripción)
DatasetImportJobError	El número de llamadas a la API CreateDatasetImportJob que han generado un error. Dimensiones: DatasetGroupArn, DatasetArn, DatasetImportJobArn
DatasetImportJobExecutionTime	El tiempo entre la llamada a la API CreateDatasetImportJob y la finalización (o el error) de la operación. Dimensiones: DatasetGroupArn, DatasetArn, DatasetImportJobArn Unidad: segundos
DatasetSize	El tamaño de los datos importados por el trabajo de importación de conjunto de datos. Dimensiones: DatasetGroupArn, DatasetArn, DatasetImportJobArn Unidad: bytes
SolutionTrainingJobRequests	El número de llamadas correctas a la API CreateSolutionVersion. Dimensiones: SolutionArn, SolutionVersionArn
SolutionTrainingJobError	El número de llamadas a la API CreateSolutionVersion que han generado un error. Dimensiones: SolutionArn, SolutionVersionArn

Métrica	Description (Descripción)
`SolutionTrainingJobExecutionTime`	El tiempo entre la llamada a la API `CreateSolutionVersion` y la finalización (o el error) de la operación. Dimensiones: `SolutionArn, SolutionVersionArn` Unidad: segundos
`GetPersonalizedRanking`	Si una llamada a la API de GetPersonalizedRanking se ha realizado correctamente. Usa la `sum` estadística para ver el recuento total de llamadas a la GetPersonalizedRanking API que se han realizado correctamente. Esta métrica no admite otras estadísticas. Dimensión: `CampaignArn`
`GetPersonalizedRanking4xxErrors`	El número de llamadas a la API `GetPersonalizedRanking` que han devuelto un código de respuesta HTTP 4xx. Dimensión: `CampaignArn`
`GetPersonalizedRanking5xxErrors`	El número de llamadas a la API `GetPersonalizedRanking` que han devuelto un código de respuesta HTTP 5xx. Dimensión: `CampaignArn`
`GetPersonalizedRankingLatency`	El tiempo transcurrido entre la recepción de la llamada a la API `GetPersonalizedRanking` y el envío de las recomendaciones (excluye los errores 4xx y 5xx). Dimensión: `CampaignArn` Unidad: milisegundos

Métrica	Description (Descripción)
GetRecommendations	Si una llamada a la API de GetRecommendations se realiza correctamente. Usa la sum estadística para ver el recuento total de llamadas a la GetRecommendations API que se han realizado correctamente. Esta métrica no admite otras estadísticas. Dimensión: CampaignArn
GetRecommendations 4xxErrors	El número de llamadas a la API GetRecommendations que han devuelto un código de respuesta HTTP 4xx. Dimensión: CampaignArn
GetRecommendations5xxErrors	El número de llamadas a la API GetRecommendations que han devuelto un código de respuesta HTTP 5xx. Dimensión: CampaignArn
GetRecommendationsLatency	El tiempo transcurrido entre la recepción de la llamada a la API GetRecommendations y el envío de las recomendaciones (excluye los errores 4xx y 5xx). Dimensión: CampaignArn Unidad: milisegundos
PutEventsRequests	El número de llamadas correctas a la API PutEvents. Dimensión: DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents4xxErrors	El número de llamadas a la API PutEvents que han devuelto un código de respuesta HTTP 4xx. Dimensión: DatasetGroupArn, DatasetArn, EventTrackerArn

Métrica	Description (Descripción)
PutEvents5xxErrors	El número de llamadas a la API PutEvents que han devuelto un código de respuesta HTTP 5xx. Dimensión: DatasetGroupArn, DatasetArn, EventTrackerArn
PutEventLatency	El tiempo necesario para la finalización de la llamada a la API PutEvents (excluye los errores 4xx y 5xx). Dimensión: DatasetGroupArn, DatasetArn, EventTrackerArn Unidad: milisegundos
PutItemsRequests	El número de llamadas correctas a la API PutItems. Dimensión: DatasetGroupArn, DatasetArn
PutItems4xxErrors	El número de llamadas a la API PutItems que han devuelto un código de respuesta HTTP 4xx. Dimensión: DatasetGroupArn, DatasetArn
PutItems5xxErrors	El número de llamadas a la API PutItems que han devuelto un código de respuesta HTTP 5xx. Dimensión: DatasetGroupArn, DatasetArn
PutItemsLatency	El tiempo necesario para la finalización de la llamada a la API PutItems (excluye los errores 4xx y 5xx). Dimensión: DatasetGroupArn, DatasetArn Unidad: milisegundos
PutUsersRequests	El número de llamadas correctas a la API PutUsers. Dimensión: DatasetGroupArn, DatasetArn

Métrica	Description (Descripción)
`PutUsers4xxErrors`	El número de llamadas a la API `PutUsers` que han devuelto un código de respuesta HTTP 4xx. Dimensión: `DatasetGroupArn, DatasetArn`
`PutUsers5xxErrors`	El número de llamadas a la API `PutUsers` que han devuelto un código de respuesta HTTP 5xx. Dimensión: `DatasetGroupArn, DatasetArn`
`PutUsersLatency`	El tiempo necesario para la finalización de la llamada a la API `PutUsers` (excluye los errores 4xx y 5xx). Dimensión: `DatasetGroupArn, DatasetArn` Unidad: milisegundos

Registro de llamadas a la API Amazon Personalize con AWS CloudTrail

Amazon Personalize está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amazon Personalize. CloudTrail captura un subconjunto de llamadas a la API para Amazon Personalize como eventos, incluidas las llamadas desde la consola Amazon Personalize y desde las llamadas de código a las API Amazon Personalize. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de Amazon Personalize. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud que se realizó a Amazon Personalize, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y habilitarlo, consulte la Guía del AWS CloudTrail usuario.

Amazon Personalize la información en CloudTrail

CloudTrail está activado en su AWS cuenta al crear la cuenta. Cuando se produce una actividad de eventos admitida en Amazon Personalize, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulta Cómo ver eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de Amazon Personalize, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de . La ruta registra los eventos de todas las regiones de la AWS partición y entrega los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte lo siguiente:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recepción de archivos de CloudTrail registro de varias regiones y recepción de archivos de CloudTrail registro de varias cuentas

Amazon Personalize permite registrar cada acción (operación de API) como un evento en los archivos de CloudTrail registro. Para obtener más información, consulte Acciones.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento userIdentity de CloudTrail .

Ejemplo: Entradas del archivo de registro de Amazon Personalize

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro con las acciones de la operación de la ListDatasetGroups API. Tenga en cuenta que, dado que la operación de la ListDatasetGroups API es una acción que no cambia de estado, la `responseElements` respuesta es nula. Para obtener más información sobre el cuerpo de los CloudTrail registros, consulta el contenido de los CloudTrail registros.

```
{
    "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"
}
```

Validación de conformidad para Amazon Personalize

Third-party los auditores evalúan la seguridad y la conformidad de Amazon Personalize como parte de varios programas de AWS conformidad. Esto incluye SOC, PCI, HIPAA y otros.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte los AWS servicios incluidos en el ámbito de aplicación por programa de conformidad y . Para obtener información general, consulte los programas de AWS cumplimiento, los programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descargar informes en AWS Artifact .

Su responsabilidad de conformidad al utilizar Amazon Personalize se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa, y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido sobre : estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre cómo diseñar una arquitectura para cumplir con la HIPAA: descubra cómo puede AWS utilizarla para ejecutar cargas de trabajo confidenciales reguladas por la. U.S Ley de Portabilidad y Responsabilidad de Seguros Médicos de EE. UU (HIPAA).
- AWS recursos AWS de cumplimiento
- Evaluación de los recursos con las reglas de la Guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub CSPM— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia de Amazon Personalize

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de

disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Amazon Personalize aprovecha la infraestructura AWS global para la resiliencia de los datos. Al crear un recurso de Amazon Personalize en una AWS región, Amazon Personalize gestiona la resiliencia y la redundancia de datos del recurso en varias zonas de disponibilidad. Para obtener una lista de AWS las regiones en las que puede crear recursos de Amazon Personalize, consulte AWS las regiones y puntos de enlace en la Referencia general de Amazon Web Services. Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte la infraestructura AWS global.

Seguridad de la infraestructura en Amazon Personalize

Como servicio gestionado, Amazon Personalize está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a Amazon Personalize a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Paquetes de cifrado con perfecto secreto directo (PFS), como el DHE (Ephemeral) o el ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Amazon Personalize e interconecta los puntos de enlace de VPC (AWS PrivateLink)

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y Amazon Personalize. Esta conexión permite a Amazon Personalize comunicarse con sus recursos en su VPC sin pasar por la red pública de Internet.

Amazon VPC es una Servicio de AWS que se utiliza para lanzar AWS recursos en una nube privada virtual (VPC) o en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Con los puntos de enlace de la VPC, la AWS red gestiona el enrutamiento entre su VPC y. Servicios de AWS

Para conectar su VPC a Amazon SNS, debe definir un punto de conexión de la VPC de tipo interfaz para Amazon Personalize. Un punto de conexión de interfaz es una interfaz de red elástica con una dirección IP privada que actúa como punto de entrada para el tráfico dirigido a un Servicio de AWS compatible. El punto de conexión proporciona una conectividad de confianza y escalable a Amazon Personalize. No requiere una puerta de enlace de Internet, una instancia de traducción de direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte Qué es Amazon VPC en la Guía del usuario de Amazon VPC.

Los puntos finales de la interfaz VPC están habilitados por. AWS PrivateLink Esta AWS tecnología permite la comunicación privada entre Servicios de AWS ellos mediante una interface de red elástica con direcciones IP privadas.

> **Note**
>
> Todos los puntos de enlace del Estándar Federal de Procesamiento de Información (FIPS) de Amazon Personalize son compatibles con. AWS PrivateLink

Temas

- Creación de un punto de conexión de la VPC de interfaz para Amazon Personalize
- Creación de una política de puntos de conexión de VPC para Amazon Personalize

Creación de un punto de conexión de la VPC de interfaz para Amazon Personalize

Puede crear un punto de enlace de VPC para el servicio Amazon Personalize con la consola Amazon VPC o con (). AWS Command Line Interface AWS CLI Para obtener más información, consulte Acceder a un AWS servicio mediante un punto de enlace de VPC de interfaz en la Guía del usuario de Amazon VPC.

Para crear un punto de conexión de VPC para Amazon Personalize, elija una de las siguientes opciones para el servicio: Puede elegir crear un punto de conexión ipv4, ipv6 o de pila doble.

- com.amazonaws. *region*.personalizar
- com.amazonaws. *region*.personalize-events
- com.amazonaws. *region*.personalize-runtime

Si habilita DNS privado para el punto de conexión, puede realizar solicitudes a la API para Amazon Personalize usando su nombre de DNS predeterminado para la región, por ejemplo, `personalize.us-east-1.api.aws`.

Creación de una política de puntos de conexión de VPC para Amazon Personalize

Puede asociar una política de puntos de conexión con su punto de conexión de VPC que controla el acceso a Amazon Personalize. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de la VPC en la guía del usuario de Amazon VPC.

Ejemplo: Política de punto de conexión de VPC que permite todas las acciones de Amazon Personalize y las acciones de passRole

Cuando se asocia a un punto de conexión, esta política concede acceso a todas las acciones de Amazon Personalize y las acciones de passRole.

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:*",
                "iam:PassRole"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Ejemplo: política de puntos finales de VPC que permite las acciones de Amazon Personalize ListDatasets

Cuando se adjunta a un punto final, esta política otorga acceso a las ListDatasets acciones de Amazon Personalize listadas.

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:ListDatasets"
            ],
            "Resource": "*"
        }
    ]
}
```

Puntos de conexión y cuotas de Amazon Personalize

En las siguientes secciones se ofrece información acerca de las directrices, las cuotas y los puntos de conexión de Amazon Personalize. Puede solicitar un aumento de la cuota para aquellas cuotas que sean ajustables a través de la consola de Service Quotas. Para obtener más información, consulte Solicitud de aumento de cuota.

Temas

- Puntos de conexión y regiones de Amazon Personalize
- Conformidad
- Cuotas de servicio
- Solicitud de aumento de cuota

Puntos de conexión y regiones de Amazon Personalize

Para ver una lista de los puntos de conexión de Amazon Personalize por región, consulte Regiones y puntos de conexión de AWS en la Referencia general de Amazon Web Services.

Conformidad

Para obtener más información sobre los programas de conformidad de Amazon Personalize, consulte Conformidad de AWS, Programas de conformidad de AWS y Servicios de AWS en ámbito por programa de conformidad.

Cuotas de servicio

Tu AWS cuenta tiene los siguientes cupos para Amazon Personalize.

Recurso	Cuota
Item interactions	
Se requiere un número mínimo de interacciones únicas entre productos para crear una versión de la solución o un recomendador.	1 000

Recurso	Cuota
Para obtener una solución personalizada, debe disponer de este número de registros después de filtrar por tipo de evento o valor del evento antes del entrenamiento.	
En el caso de User-Personalization-v Personalized-Ranking-v 2 y 2 recetas, es el número máximo de interacciones entre elementos que un modelo tiene en cuenta durante el entrenamiento.	3000 millones
Para todos los casos de uso de dominios y recetas personalizadas que no sean User-Personalization-v Personalized-Ranking-v 2 o 2, el número máximo de interacciones entre elementos que tiene en cuenta un modelo durante el entrenamiento.	500 millones (ajustable)
Número máximo de tipos de eventos distintos combinado con el número total de columnas de metadatos opcionales en un conjunto de datos de interacciones entre elementos.	10
Número máximo de columnas de metadatos, excluidos los campos reservados, en un conjunto de datos de interacciones entre elementos.	5
Número máximo de caracteres para los datos categóricos y los valores de impresión.	1 000
Cantidad máxima de datos de interacciones entre elementos masivos por trabajo de importación de conjuntos de datos con el modo de importación COMPLETA.	100 GB (aumenta a 1 TB con cualquier aumento de las interacciones entre elementos considerado por un modelo)

Recurso	Cuota
Cantidad máxima de datos de interacciones masivas de elementos por trabajo de importación de conjuntos de datos con el modo de importación INCREMENTAL.	1 GB
Número mínimo de registros de interacciones entre elementos por trabajo de importación de conjuntos de datos con el modo de importación COMPLETO o INCREMENTAL.	1 000
Users	
Se requiere un número mínimo de usuarios únicos en los datos de interacciones entre elementos, con un mínimo de 2 interacciones entre elementos cada uno, para crear un recomendador de dominios o una versión de solución personalizada.	25
Porcentaje mínimo del total de usuarios que deben tener al menos 2 interacciones entre elementos para poder crear una versión de recomendación de dominios o una versión de solución personalizada.	1 por ciento
Número máximo de campos de metadatos para un conjunto de datos de usuarios.	25
Número máximo de caracteres para los valores de datos USER_ID.	256
Número máximo de caracteres para los valores de datos categóricos.	1.000 caracteres
Cantidad máxima de datos de usuario masivos por trabajo de importación de conjuntos de datos con el modo de importación COMPLETA.	100 GB

Recurso	Cuota
Cantidad máxima de datos de usuario masivos por trabajo de importación de conjuntos de datos con el modo de importación INCREMENTAL.	1 GB
Items	
Para User-Personalization-v 2 o Personalized-Ranking-v 2, la cantidad máxima de elementos que un modelo considera durante el entrenamiento. Estos elementos provienen del conjunto de datos de elementos y de interacciones de elemento.	5 millones
Para todos los casos de uso de dominios y recetas personalizadas que no sean User-Personalization-v Personalized-Ranking-v 2 y 2, el número máximo de elementos que un modelo tiene en cuenta durante el entrenamiento y la generación de recomendaciones.	750 000
Número máximo de campos de metadatos para un conjunto de datos de Items.	100
Número máximo de caracteres para los valores de datos ITEM_ID.	256
Número máximo de caracteres para valores de datos de cadena categóricos y no categóricos.	1.000 caracteres
Número máximo de campos textuales para un conjunto de datos de Items.	1
Número máximo de caracteres para los valores de datos textuales en los idiomas chino y japonés.	7.000 caracteres

Recurso	Cuota
Número máximo de caracteres para los valores de datos textuales de todos los demás idiomas.	20 000 caracteres
Cantidad máxima de datos de elementos masivos por trabajo de importación de conjuntos de datos con el modo de importación MASIVA.	100 GB
Cantidad máxima de datos de elementos masivos por trabajo de importación de conjuntos de datos con el modo de importación INCREMENTAL.	1 GB
Actions	
Número máximo de acciones que tiene en cuenta un modelo durante el entrenamiento y la generación de recomendaciones	1 000
Número máximo de campos de metadatos para un conjunto de datos de Actions.	10
Número máximo de caracteres para los valores de datos de ACTION_ID.	256
Número máximo de caracteres para los valores de datos categóricos.	1.000 caracteres
Cantidad máxima de datos de acciones masivas por trabajo de importación de conjuntos de datos con el modo de importación MASIVA.	100 GB
Cantidad máxima de datos de acciones masivas por trabajo de importación de conjuntos de datos con el modo de importación INCREMENTAL.	1 GB

Recurso	Cuota
Action interactions	
Número máximo de interacciones que se consideran en un modelo durante el entrenamiento	500 millones
Número máximo de columnas de metadatos, excluidos los campos reservados, en un conjunto de datos de interacciones entre acciones.	5
Cantidad máxima de datos de interacciones masivas por trabajo de importación de conjuntos de datos con el modo de importación COMPLETA.	100 GB (aumenta a 1 TB si el modelo considera el aumento de las interacciones entre elementos de acción)
Cantidad máxima de datos de interacciones masivas por trabajo de importación de conjuntos de datos con el modo de importación INCREMENTAL.	1 GB
Individual record import APIs	
Tasa máxima de `PutEvents` solicitudes por grupo de conjuntos de datos.	1000/segundo
Número máximo de eventos en una llamada `PutEvents` .	10
Tamaño máximo de un evento.	10 KB
Tasa máxima de `PutActionInteractions` solicitudes por grupo de conjuntos de datos.	1000/segundo

Recurso	Cuota
Número máximo de eventos de interacción entre acciones en una `PutActionInteractions` llamada.	10
Tamaño máximo de un evento de interacción entre acciones.	10 KB
Tasa máxima de `PutItems` solicitudes por grupo de conjuntos de datos.	10 por segundo
Número máximo de artículos en una `PutItems` llamada.	10
Tasa máxima de `PutUsers` solicitudes por grupo de conjuntos de datos.	10 por segundo
Número máximo de usuarios en una `PutUsers` llamada.	10
Tasa máxima de `PutActions` solicitudes por grupo de conjuntos de datos.	10 por segundo
Número máximo de usuarios en una `PutActions` llamada.	10
Legacy recipes	
Cantidad máxima de datos combinados para los conjuntos de datos de usuarios y elementos de las recetas de HRNN-Metadata y HRNN-Coldstart.	5 GB
Número máximo de elementos de arranque en frío que admite la receta HRNN-ColdStart para entrenar un modelo (crear una versión de solución).	80.000

Recurso	Cuota
Número mínimo de elementos de arranque en frío que requiere la receta HRNN-ColdStart para entrenar un modelo (crear una versión de solución).	100
Filters	
Número total de filtros por grupo de conjuntos de datos.	30 (ajustable)
Número máximo de campos de conjunto de datos distintos para un filtro.	10
Número total de campos de conjuntos de datos distintos en todos los filtros de un grupo de conjuntos de datos.	20
Número máximo de interacciones entre elementos por usuario y por tipo de evento considerado por un filtro.	100 interacciones (ajustable)
Número máximo de interacciones de acción por usuario y por tipo de evento considerado por un filtro.	300 interacciones de acción (ajustables)
GetRecommendations / GetPersonalizedRanking / GetActionRecommendations requests	
Tasa máxima de `GetRecommendations` transacciones `GetActionRecommendations` y `GetPersonalizedRanking` solicitudes.	2500/s
Número máximo de solicitudes `GetRecommendations` por segundo y por campaña.	500/s

Recurso	Cuota
Número máximo de solicitudes `GetActionRecommendations` por segundo y por campaña.	500/s
Número máximo de solicitudes `GetPersonalizedRanking` por segundo y por campaña.	500/seg.
Número máximo de columnas de metadatos por solicitud`GetRecommendations` . `GetPersonalizedRanking`	10
Número máximo de resultados de recomendación para una `GetRecommendation` solicitud sin metadatos.	500
Número máximo de resultados de recomendación para una `GetRecommendation` solicitud con metadatos.	50
Número máximo de elementos que se pueden clasificar en una `GetPersonalizedRanking` solicitud sin metadatos.	500
Número máximo de elementos para clasificar en una `GetPersonalizedRanking` solicitud con metadatos.	50
Metric attribution quotas	
Número máximo de métricas para una atribución de métrica	10
Número máximo de fuentes únicas de atribución de eventos	100
Batch inference jobs	

Recurso	Cuota
Número máximo de archivos de entrada para un trabajo de inferencia por lotes.	1 000
Tamaño máximo de la entrada de un trabajo de inferencia por lotes.	1 GB
Número máximo de registros por archivo de entrada para un trabajo de inferencia por lotes sin temas.	50 millones
Número máximo de registros por archivo de entrada para un trabajo de inferencia por lotes con temas.	100
Batch segment jobs	
Número máximo de archivos de entrada para un trabajo de segmento por lotes.	1 000
Tamaño máximo de la entrada del trabajo segmentado por lotes.	1 GB
Número máximo de consultas por archivo de entrada para la receta de Item-Afinidad.	500
Número máximo de consultas por archivo de entrada para Item-Attribute-Affinity la receta.	10
Número máximo de usuarios por segmento	5 millones
Data deletion jobs	
Número máximo de trabajos de eliminación de datos para un grupo de conjuntos de datos con el estado PENDIENTE.	5 (ajustable)
Tamaño total máximo del archivo o archivos de entrada para la eliminación de datos	100 MB

Tu AWS cuenta tiene las siguientes cuotas para cada región.

Recurso	Cuota
Número total de esquemas activos.	500
Número total de grupos de conjuntos de datos activos.	5 (ajustable)
Número total de trabajos de importación de conjuntos de datos pendientes o en curso.	5
Número total de trabajos de inferencia por lotes pendientes o en curso.	5 (ajustable)
Número total de trabajos de segmentos por lotes pendientes o en curso.	5
Número total de versiones de la solución pendientes o en curso.	20 (ajustable)

Cada grupo de conjuntos de datos tiene las siguientes cuotas.

Recurso	Cuota
Número total de soluciones activas.	10 (ajustable)
Número total de campañas activas.	5 (ajustable)
Número total de recomendadores.	5
Número total de filtros.	30 (ajustable)
Número total de campos de conjuntos de datos distintos en todos los filtros.	20
Número total de trabajos de eliminación de datos para un grupo de conjuntos de datos con el estado PENDING.	5

Solicitud de aumento de cuota

Puede solicitar un aumento de la cuota para aquellas cuotas que sean ajustables a través de la consola de Service Quotas. Las siguientes cuotas de Amazon Personalize son ajustables.

- Número máximo de interacciones de elementos que se consideran en un modelo durante el entrenamiento
- Campañas activas por grupo de conjuntos de datos
- Grupos de conjuntos de datos activos
- Filtros activos por grupo de conjuntos de datos
- Soluciones activas por grupo de conjuntos de datos
- Cantidad de datos por importación incremental
- Número máximo de interacciones de elementos por usuario y por tipo de evento considerado por un filtro
- Número total de trabajos de inferencia por lotes pendientes o en curso
- Número total de versiones de la solución pendientes o en curso
- Tasa máxima de solicitudes `PutEvents` o `PutActionInteraction`

Para solicitar un aumento de las cuotas, utilice la Consola de Service Quotas y siga los pasos de la sección Solicitud de un aumento de cuota de la Guía del usuario de Service Quotas.

Referencia de la API

Esta sección proporciona documentación sobre las operaciones de API de Amazon Personalize. Para ver una lista de los puntos de conexión de Amazon Personalize por región, consulte Regiones y puntos de conexión de AWS en la Referencia general de AWS .

Temas

- Acciones
- Data Types
- Tipos de errores comunes
- Parámetros comunes

Acciones

Amazon Personalize admite las siguientes acciones:

- 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

Eventos de Amazon Personalize admite las siguientes acciones:

- PutActionInteractions
- PutActions
- PutEvents

- PutItems
- PutUsers

Versión ejecutable de Amazon Personalize admite las siguientes acciones:

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

Amazon Personalize

Amazon Personalize admite las siguientes acciones:

- 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

Servicio: Amazon Personalize

Genera recomendaciones por lotes basadas en una lista de elementos o de usuarios almacenados en Amazon S3 y exporta las recomendaciones a un bucket de Amazon S3.

Para generar recomendaciones por lotes, especifique el ARN de una versión de la solución y un URI de Amazon S3 para los datos de entrada y salida. Para la personalización de usuarios, elementos populares y soluciones de clasificación personalizadas, el trabajo de inferencia por lotes genera una lista de elementos recomendados para cada ID de usuario del archivo de entrada. Para las soluciones de elementos relacionadas, el trabajo genera una lista de elementos recomendados para cada ID de elemento del archivo de entrada.

Para obtener más información, consulte Creación de un trabajo de inferencia por lotes.

Si utilizas la Similar-Items receta, Amazon Personalize puede añadir temas descriptivos a las recomendaciones de lotes. Para generar temas, establezca el modo del trabajo en `THEME_GENERATION` y especifique el nombre del campo que contiene los nombres de los elementos en los datos de entrada.

Para obtener más información sobre la generación de temas, consulte Recomendaciones por lotes con temas del Generador de contenidos.

No puedes obtener recomendaciones por lotes con las Next-Best-Action recetas Trending-Now o.

Sintaxis de la solicitud

```
{
   "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"
         }
      }
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

batchInferenceJobConfig

Los detalles de configuración de un trabajo de inferencia por lotes.

Tipo: objeto BatchInferenceJobConfig

Obligatorio: no

batchInferenceJobMode

Modo del trabajo de inferencia por lotes. Para generar temas descriptivos para grupos de elementos similares, establezca el modo de trabajo en `THEME_GENERATION`. Si no quiere generar temas, use el valor predeterminado `BATCH_INFERENCE`.

Al recibir recomendaciones por lotes con temas, incurrirá en costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Tipo: cadena

Valores válidos: `BATCH_INFERENCE | THEME_GENERATION`

Obligatorio: no

filterArn

El ARN del filtro que se va a aplicar al trabajo de inferencia por lotes. Para obtener más información, consulte Filtrado de recomendaciones de lote.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

jobInput

La ruta de Amazon S3 que lleva al archivo de entrada en el que basar sus recomendaciones. El material de entrada debe estar en formato JSON.

Tipo: objeto BatchInferenceJobInput

Obligatorio: sí

jobName

El nombre del trabajo de inferencia por lotes que se va a crear.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

jobOutput

La ruta al bucket de Amazon S3 donde se almacenará el resultado del trabajo.

Tipo: objeto BatchInferenceJobOutput

Obligatorio: sí

numResults

El número de recomendaciones que se van a recuperar.

Tipo: entero

Obligatorio: no

roleArn

El ARN del rol de Amazon Identity and Access Management que tiene permisos de lectura y escritura en los buckets de Amazon S3 de entrada y salida, respectivamente.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: sí

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de la solución que se va a utilizar para generar las recomendaciones de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Una lista de etiquetas para aplicar al trabajo de inferencia por lotes.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

themeGenerationConfig

Para los trabajos de generación de temas, especifique el nombre de la columna del conjunto de datos de elementos que contiene el nombre de cada elemento.

Tipo: objeto ThemeGenerationConfig

Obligatorio: no

Sintaxis de la respuesta

```
{
   "batchInferenceJobArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchInferenceJobArn

El ARN del trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateBatchSegmentJob

Servicio: Amazon Personalize

Crea un trabajo de segmento por lotes. La operación puede gestionar hasta 50 millones de registros y el archivo de entrada debe estar en formato JSON. Para obtener más información, consulte Obtención de recomendaciones por lotes y segmentos de usuario.

Sintaxis de la solicitud

```
{
   "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"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

filterArn

El ARN del filtro que se va a aplicar al trabajo de segmentación por lotes. Para obtener más información, consulte Filtrado de recomendaciones de lote.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

jobInput

La ruta de Amazon S3 para los datos de entrada utilizados para generar el trabajo de segmentación por lotes.

Tipo: objeto BatchSegmentJobInput

Obligatorio: sí

jobName

El nombre del trabajo de segmentación por lotes que se va a crear.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

jobOutput

La ruta de Amazon S3 del bucket donde se almacenará el resultado del trabajo.

Tipo: objeto BatchSegmentJobOutput

Obligatorio: sí

numResults

El número de usuarios previsto que generará el trabajo de segmentación por lotes para cada línea de datos de entrada. El número máximo de usuarios por segmento es de 5 millones.

Tipo: entero

Obligatorio: no

roleArn

El ARN del rol de Amazon Identity and Access Management que tiene permisos de lectura y escritura en los buckets de Amazon S3 de entrada y salida, respectivamente.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: sí

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de la solución que desea que el trabajo de segmentación por lotes use para generar segmentos por lote.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Una lista de etiquetas para aplicar al trabajo de segmentación por lotes.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "batchSegmentJobArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchSegmentJobArn

El ARN del trabajo de segmentación por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateCampaign

Servicio: Amazon Personalize

> **Important**
>
> Generará costos de campaña mientras esté activa. Para evitar costos innecesarios, asegúrese de eliminar la campaña cuando haya terminado. Para obtener información sobre los costos de campaña, consulte Precios de Amazon Personalize.

Crea una campaña que implementa una versión de solución. Cuando un cliente llama a las GetPersonalizedRanking API GetRecommendations y, se especifica una campaña en la solicitud.

TPS mínimo aprovisionado y Auto-Scaling

> **Important**
>
> Un valor elevado de `minProvisionedTPS` aumentará los costos. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realiza un seguimiento de tu uso mediante CloudWatch las métricas de Amazon y `minProvisionedTPS` auméntalas según sea necesario.

Al crear una campaña de Amazon Personalize, puede especificar el número mínimo de transacciones aprovisionadas por segundo (`minProvisionedTPS`) para la campaña. Este es el rendimiento de transacciones de línea base para la campaña aprovisionada por Amazon Personalize. Establece el cargo de facturación mínimo de la campaña mientras esté activa. Una transacción es una solicitud `GetRecommendations` o `GetPersonalizedRanking` simple. El valor predeterminado de `minProvisionedTPS` es 1.

Si su TPS aumenta más de `minProvisionedTPS`, Amazon Personalize escala automáticamente la capacidad aprovisionada hacia arriba y hacia abajo, pero nunca por debajo de `minProvisionedTPS`. Se produce un breve retraso durante el aumento de la capacidad que podría provocar la pérdida de transacciones. Cuando se reduce el tráfico, la capacidad vuelve al valor de `minProvisionedTPS`.

Se le cobrará por el valor de TPS mínimo aprovisionado o, si sus solicitudes superan el valor de `minProvisionedTPS`, el TPS real. El TPS real es el número total de solicitudes de recomendación

que realiza. Te recomendamos empezar con un nivel mínimo`minProvisionedTPS`, realizar un seguimiento de tu uso con CloudWatch las métricas de Amazon y, a continuación, aumentar las métricas `minProvisionedTPS` según sea necesario.

Para obtener más información sobre los costos de las campañas, consulte Precios de Amazon Personalize.

Estado

Una campaña puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Para conocer el estado de la campaña, llama DescribeCampaign.

> **Note**
>
> Espera a que el `status` de la campaña sea `ACTIVE` antes de pedir recomendaciones a la campaña.

API relacionadas

- ListCampaigns
- DescribeCampaign
- UpdateCampaign
- DeleteCampaign

Sintaxis de la solicitud

```
{
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "rankingInfluence": {
         "string" : number
```

```
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
   "name": "string",
   "solutionVersionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignConfig

Los detalles de la configuración de una campaña.

Tipo: objeto CampaignConfig

Obligatorio: no

minProvisionedTPS

Especifica el número mínimo de transacciones aprovisionadas (recomendaciones) por segundo solicitadas que Amazon Personalize va a admitir. Un valor elevado de `minProvisionedTPS` aumentará su factura. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realiza un seguimiento de tu uso con CloudWatch las métricas de Amazon y `minProvisionedTPS` auméntalas según sea necesario.

Tipo: entero

Rango válido: valor mínimo de 1.

Obligatorio: no

name

Un nombre para la nueva campaña. El nombre de la campaña debe ser único en la cuenta.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

solutionVersionArn

Nombre de recurso de Amazon (ARN) del modelo entrenado que se implementará con la campaña. Para especificar la versión más reciente de la solución, especifique el ARN de la solución en formato `SolutionArn/$LATEST`. Debes usar este formato si lo `syncWithLatestSolutionVersion` configuraste `True` en CampaignConfig.

Para implementar un modelo que no sea la última versión de solución, especifique el ARN de la versión de solución.

Para obtener más información sobre las actualizaciones de campaña automáticas, consulte Habilitación de las actualizaciones de campaña automáticas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Lista de etiquetas que se aplican a la campaña.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "campaignArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateDataDeletionJob

Servicio: Amazon Personalize

Crea un trabajo por lotes que elimina por lotes todas las referencias a usuarios específicos de un grupo de conjuntos de datos de Amazon Personalize. Especifique los usuarios que se eliminarán en un archivo CSV de ID de usuario en un bucket de Amazon S3. Una vez finalizado un trabajo, Amazon Personalize ya no utiliza los datos de los usuarios para el entrenamiento y deja de tener en cuenta a los usuarios al generar segmentos de usuarios. Para obtener más información sobre cómo crear un trabajo de eliminación de datos, consulte Eliminación de usuarios.

- El archivo de entrada debe ser un archivo CSV con una sola columna USER_ID que enumere los ID de los usuarios. Para obtener más información sobre cómo preparar el archivo CSV, consulte Cómo preparar el archivo de eliminación de datos y cargarlo en Amazon S3.
- Para conceder permiso a Amazon Personalize para acceder al archivo CSV de entrada de los ID de usuario, debe especificar un rol de servicio de IAM que tenga permiso para leer el origen de datos. Este rol necesita los permisos `GetObject` y `ListBucket` para el bucket y el contenido. Estos permisos son los mismos que en la importación de datos. Para obtener información sobre cómo conceder acceso a su bucket de Amazon S3, consulte Concesión de acceso a los recursos de Amazon S3 para Amazon Personalize.

Tras crear un trabajo, es posible que se tarde hasta un día en eliminar todas las referencias a los usuarios de los conjuntos de datos y modelos. Hasta que se complete el trabajo, Amazon Personalize seguirá utilizando los datos durante el entrenamiento. Y si utiliza una receta de segmentación de usuarios, es posible que los usuarios aparezcan en segmentos de usuarios.

Estado

Un trabajo de eliminación de datos puede tener uno de los siguientes estados:

- PENDING > IN_PROGRESS > COMPLETED -o- FAILED

Para obtener el estado del trabajo de eliminación de datos, llame a la operación de la DescribeDataDeletionJob API y especifique el nombre de recurso de Amazon (ARN) del trabajo. Si el estado es FAILED, la respuesta incluye una clave `failureReason` que describe el motivo del error en el trabajo.

API relacionadas

- ListDataDeletionJobs
- DescribeDataDeletionJob

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "dataSource": {
      "dataLocation": "string"
   },
   "jobName": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

Nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que tiene los conjuntos de datos cuyos registros desea eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

dataSource

Bucket de Amazon S3 que contiene la lista de ID de los usuarios que se eliminarán.

Tipo: objeto DataSource

Obligatorio: sí

jobName

Nombre del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

roleArn

Nombre de recurso de Amazon (ARN) del rol de IAM que tiene permisos para leer el origen de datos de Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: sí

tags

Lista de etiquetas que se aplican al trabajo de eliminación de datos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "dataDeletionJobArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

dataDeletionJobArn

Nombre de recurso de Amazon (ARN) del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateDataset

Servicio: Amazon Personalize

Crea un conjunto de datos vacío y lo agrega al grupo de conjuntos de datos especificado. CreateDatasetImportJobUtilízalo para importar tus datos de entrenamiento a un conjunto de datos.

Existen 5 tipos de conjuntos de datos:

- Interacciones de elementos
- Items
- Users
- Interacciones de acción
- Acciones

Cada conjunto de datos tiene asociado un esquema con los tipos de campo necesarios. Solo es necesario el conjunto de datos `Item interactions` para entrenar un modelo (también denominado creación de una solución).

Un conjunto de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Para obtener el estado del conjunto de datos, llama DescribeDataset.

API relacionadas

- CreateDatasetGroup
- ListDatasets
- DescribeDataset
- DeleteDataset

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "datasetType": "string",
```

```
   "name": "string",
   "schemaArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjunto de datos al que se va a agregar el conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

datasetType

El tipo de conjunto de datos.

Uno de los siguientes valores (no se distingue entre mayúsculas y minúsculas):

- Interacciones
- Items
- Users
- Acciones
- Action_Interactions

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `^[A-Za-z_]+$`

Obligatorio: sí

name

Nombre del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

schemaArn

El ARN del esquema que se debe asociar al conjunto de datos. El esquema define los campos de un conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Lista de etiquetas que se aplican al conjunto de datos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetArn

El ARN del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateDatasetExportJob

Servicio: Amazon Personalize

Crea un trabajo que exporta datos desde su conjunto de datos a un bucket de Amazon S3. Para permitir que Amazon Personalize exporte los datos de entrenamiento, debe especificar un rol de IAM vinculado al servicio que otorgue a Amazon Personalize permisos de `PutObject` para su bucket de Amazon S3. Para obtener más información, consulte Exportación de un conjunto de datos en la guía para desarrolladores de Amazon Personalize.

Estado

Un trabajo de exportación de conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Para obtener el estado del trabajo de exportación DescribeDatasetExportJob, llame y especifique el nombre de recurso de Amazon (ARN) del trabajo de exportación del conjunto de datos. La exportación del conjunto de datos finaliza cuando el estado es ACTIVE. Si el estado muestra CREATE FAILED, la respuesta incluye una clave `failureReason` que describe el motivo del error en el trabajo.

Sintaxis de la solicitud

```
{
   "datasetArn": "string",
   "ingestionMode": "string",
   "jobName": "string",
   "jobOutput": {
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que contiene los datos que se van a exportar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

ingestionMode

Los datos que se van a exportar, en función de cómo los haya importado. Puedes optar por exportar solo `BULK` los datos que hayas importado mediante un trabajo de importación de conjuntos de datos, solo `PUT` los datos que hayas importado de forma incremental (mediante la consola PutUsers y PutItems las operaciones) o `ALL` para ambos tipos. PutEvents El valor predeterminado es `PUT`.

Tipo: cadena

Valores válidos: `BULK | PUT | ALL`

Obligatorio: no

jobName

Nombre para el trabajo de exportación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

jobOutput

La ruta al bucket de Amazon S3 donde se almacena el resultado del trabajo.

Tipo: objeto DatasetExportJobOutput

Obligatorio: sí

roleArn

El nombre de recurso de Amazon (ARN) del rol de servicio de IAM que tiene permisos para añadir datos al bucket de Amazon S3 de salida.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: sí

tags

Lista de etiquetas que se aplican al trabajo de exportación del conjunto de datos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetExportJobArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetExportJobArn

El nombre de recurso de Amazon (ARN) del trabajo de exportación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateDatasetGroup

Servicio: Amazon Personalize

Crea un grupo de conjuntos de datos vacío. Un grupo de conjuntos de datos es un contenedor para los recursos de Amazon Personalize. Un grupo de conjuntos de datos puede contener como máximo tres conjuntos de datos, uno para cada tipo de conjunto de datos:

- Interacciones de elementos
- Items
- Users
- Acciones
- Interacciones de acción

Un grupo de conjuntos de datos puede ser un grupo de conjuntos de datos de dominio, en el que se especifica un dominio y se utilizan recursos preconfigurados, como recomendaciones, o un grupo de conjuntos de datos personalizados, en el que se utilizan recursos personalizados, como una solución con una versión de solución, que se implementa con una campaña. Si empieza con un grupo de conjuntos de datos de dominio, puede añadir recursos personalizados, como soluciones y versiones de soluciones entrenadas con recetas para casos de uso personalizados e implementadas con campañas.

Un grupo de conjuntos de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING

Para obtener el estado del grupo de conjuntos de datos, llame a DescribeDatasetGroup. Si el estado muestra CREATE FAILED, la respuesta incluye una clave `failureReason` que describe el motivo del error en la creación.

> **Note**
>
> Debe esperar a que el valor `status` del grupo de conjuntos de datos sea `ACTIVE` antes de agregar un conjunto de datos al grupo.

Puede especificar una clave AWS Key Management Service (KMS) para cifrar los conjuntos de datos del grupo. Si especifica una clave de KMS, también debe incluir un rol (de IAM) AWS Identity and Access Management que tenga permiso para acceder a la clave.

API que requieren un ARN de grupo de conjuntos de datos en la solicitud

- CreateDataset
- CreateEventTracker
- CreateSolution

API relacionadas

- ListDatasetGroups
- DescribeDatasetGroup
- DeleteDatasetGroup

Sintaxis de la solicitud

```
{
   "domain": "string",
   "kmsKeyArn": "string",
   "name": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

domain

El dominio del grupo de conjuntos de datos. Especifique un dominio para crear un grupo de conjuntos de datos de dominio. El dominio que especifique determina los esquemas

predeterminados de los conjuntos de datos y los casos de uso disponibles para los recomendadores. Si no especifica un dominio, creará un grupo de conjuntos de datos personalizados con versiones de soluciones que puede implementar con una campaña.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

kmsKeyArn

El nombre de recurso de Amazon (ARN) de una clave AWS Key Management Service (KMS) utilizada para cifrar los conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 2048 caracteres.

Patrón: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatorio: no

name

El nombre del nuevo grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

roleArn

El ARN del rol AWS Identity and Access Management (IAM) que tiene permisos para acceder a la clave (KMS). AWS Key Management Service La especificación de un rol de IAM solo es válida cuando también se especifica una clave de KMS.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

tags

Lista de etiquetas que se aplican al grupo de conjuntos de datos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetGroupArn": "string",
   "domain": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del nuevo grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

domain

El dominio para el nuevo grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateDatasetImportJob

Servicio: Amazon Personalize

Crea un trabajo que importa datos de entrenamiento de un origen de datos (bucket de Amazon S3) a un conjunto de datos de Amazon Personalize. Para permitir que Amazon Personalize importe los datos de entrenamiento, debe especificar un rol de servicio de IAM que tenga permiso para leer el origen de datos, ya que Amazon Personalize hace una copia de sus datos y los procesa internamente. Para obtener información sobre cómo conceder acceso a su bucket de Amazon S3, consulte Concesión de acceso a los recursos de Amazon S3 para Amazon Personalize.

Si ya ha creado un recomendador o ha implementado una versión de solución personalizada con una campaña, la forma en que los nuevos registros masivos influyen en las recomendaciones depende del caso de uso del dominio o de la receta que utilice. Para obtener más información, consulta Influencia de los nuevos datos en las recomendaciones en tiempo real.

> **Important**
>
> De forma predeterminada, un trabajo de importación de conjuntos de datos reemplaza cualquier dato existente en el conjunto de datos que haya importado de manera masiva. Para agregar nuevos registros sin reemplazar los datos existentes, especifique INCREMENTAL para el modo de importación de la CreateDatasetImportJob operación.

Estado

Un trabajo de importación del conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING (CREACIÓN PENDIENTE) > CREATE IN_PROGRESS (CREACIÓN EN CURSO) > ACTIVE (ACTIVO) o CREATE FAILED (ERROR AL CREAR)

Para obtener el estado del trabajo de importación DescribeDatasetImportJob, llame e indique el nombre de recurso de Amazon (ARN) del trabajo de importación del conjunto de datos. La importación del conjunto de datos finaliza cuando el estado es ACTIVO. Si el estado muestra CREATE FAILED, la respuesta incluye una clave `failureReason` que describe el motivo del error en el trabajo.

> **Note**
>
> La importación lleva tiempo. Deberá esperar hasta que el estado pase a ACTIVE antes de entrenar un modelo con el conjunto de datos.

API relacionadas

- ListDatasetImportJobs
- DescribeDatasetImportJob

Sintaxis de la solicitud

```
{
   "datasetArn": "string",
   "dataSource": {
      "dataLocation": "string"
   },
   "importMode": "string",
   "jobName": "string",
   "publishAttributionMetricsToS3": boolean,
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El ARN del conjunto de datos que recibe los datos importados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

dataSource

El bucket de Amazon S3 que contiene los datos de entrenamiento que se importarán.

Tipo: objeto DataSource

Obligatorio: sí

importMode

Especifique cómo agregar los nuevos registros a un conjunto de datos existente. El modo de importación predeterminado es `FULL`. Si no ha importado registros masivos al conjunto de datos anteriormente, solo puede especificar `FULL`.

- Especifique `FULL` para que se sobrescriban todos los datos masivos existentes en el conjunto de datos. Los datos que ha importado individualmente no se reemplazan.
- Especifique `INCREMENTAL` si desea anexar los nuevos registros a los datos existentes en su conjunto de datos. Amazon Personalize reemplaza cualquier registro con el mismo identificador por el nuevo.

Tipo: cadena

Valores válidos: `FULL | INCREMENTAL`

Obligatorio: no

jobName

Nombre para el trabajo de importación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

publishAttributionMetricsToS3

Si ha creado una atribución de métricas, especifique si desea publicar las métricas de este trabajo de importación en Amazon S3

Tipo: Booleano

Obligatorio: no

roleArn

El ARN del rol de IAM que tiene permisos de lectura del origen de datos de Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

tags

Lista de etiquetas que se aplican al trabajo de importación del conjunto de datos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetImportJobArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetImportJobArn

El ARN del trabajo de importación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateEventTracker

Servicio: Amazon Personalize

Crea un rastreador de eventos que se usa al agregar datos de eventos a un grupo de conjuntos de datos específico mediante la PutEvents API.

> **Note**
>
> Solo se puede asociar un único rastreador de ventas con un grupo de conjuntos de datos. Se producirá un error si llama a `CreateEventTracker` utilizando el mismo grupo de conjuntos de datos que un rastreador de eventos existente.

Cuando creas un rastreador de eventos, la respuesta incluye un identificador de seguimiento, que pasas como parámetro cuando usas la PutEvents operación. A continuación, Amazon Personalize anexa los datos del evento al conjunto de datos de interacciones de elementos del grupo de conjuntos de datos que especifique en el rastreador de eventos.

El rastreador de eventos de puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Para obtener el estado del rastreador de eventos, llama DescribeEventTracker.

> **Note**
>
> El rastreador de eventos debe estar en estado ACTIVO antes de usar el ID de seguimiento.

API relacionadas

- ListEventTrackers
- DescribeEventTracker
- DeleteEventTracker

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "name": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que recibe los datos del evento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

name

El nombre del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

tags

Una lista de etiquetas para aplicar al rastreador de eventos.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "eventTrackerArn": "string",
   "trackingId": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

eventTrackerArn

El ARN del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

trackingId

ID del rastreador de eventos. Incluye este ID en las solicitudes a la PutEvents API.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateFilter

Servicio: Amazon Personalize

Crea un filtro de recomendaciones. Para obtener más información, consulte Filtrado de recomendaciones por lotes y segmentos de usuario.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "filterExpression": "string",
   "name": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El ARN del grupo de conjuntos de datos al que pertenecerá el filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

filterExpression

La expresión del filtro define qué elementos se incluyen en las recomendaciones o se excluyen de estas. La expresión de filtro debe seguir reglas de formato específicas. Para obtener información sobre la estructura y la sintaxis de la expresión de filtro, consulte Expresiones de filtro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 2500.

Obligatorio: sí

name

Nombre del filtro que se va a crear.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

tags

Lista de etiquetas que se aplican al filtro.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "filterArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

filterArn

El ARN del nuevo filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2

- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateMetricAttribution

Servicio: Amazon Personalize

Crea una atribución de métricas. La atribución de métricas crea informes sobre los datos que importa en Amazon Personalize. Según cómo haya importado los datos, puede ver los informes en Amazon CloudWatch o Amazon S3. Para obtener más información, consulte Medición del impacto de las recomendaciones.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "name": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El Nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de destino para la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

metrics

Una lista de atributos de métrica para la atribución de métricas. Cada atributo de métrica especifica un tipo de evento que se debe rastrear y una función. Las funciones disponibles son `SUM()` o `SAMPLECOUNT()`. Para las funciones SUM(), proporcione el tipo de conjunto de datos (interacciones o elementos) y la columna que se va a sumar como parámetro. Por ejemplo, SUM (Items.PRICE).

Tipo: matriz de objetos MetricAttribute

Miembros de la matriz: número máximo de 10 elementos.

Obligatorio: sí

metricsOutputConfig

Los detalles de configuración de salida para la atribución de métricas.

Tipo: objeto MetricAttributionOutput

Obligatorio: sí

name

Un nombre para la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "metricAttributionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metricAttributionArn

El Nombre de recurso de Amazon (ARN) para la nueva atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateRecommender

Servicio: Amazon Personalize

Crea un recomendador con la receta (un caso de uso de un grupo de conjuntos de datos de dominio) que especifique. Creas recomendadores para un grupo de conjuntos de datos de dominio y especificas el nombre de recurso de Amazon (ARN) del recomendante cuando haces una solicitud. GetRecommendations

Solicitudes de recomendación mínimas por segundo

> **Important**
>
> Un valor elevado de `minRecommendationRequestsPerSecond` aumentará su factura. Le recomendamos empezar con 1 para `minRecommendationRequestsPerSecond` (el valor predeterminado). Realiza un seguimiento de tu uso mediante CloudWatch las métricas de Amazon y auméntalos según sea necesario. `minRecommendationRequestsPerSecond`

Al crear un recomendador, puede configurar las solicitudes de recomendación mínimas por segundo de este. El número mínimo de solicitudes de recomendación por segundo (`minRecommendationRequestsPerSecond`) especifica el rendimiento básico de las solicitudes de recomendación aprovisionado por Amazon Personalize. El mínimo predeterminado RecommendationRequestsPerSecond es1. Una solicitud de recomendación es una operación `GetRecommendations` única. El rendimiento de las solicitudes se mide en solicitudes por segundo y Amazon Personalize utiliza las solicitudes por segundo para derivar las solicitudes por hora y el precio del uso del recomendador.

Si sus solicitudes por segundo aumentan más allá de `minRecommendationRequestsPerSecond`, Amazon Personalize escala y reduce automáticamente la capacidad aprovisionada, pero nunca por debajo de `minRecommendationRequestsPerSecond`. Se produce un breve retraso durante el aumento de la capacidad que podría provocar la pérdida de solicitudes.

Tu factura es el mayor entre el número mínimo de solicitudes por hora (basado en el mínimoRecommendationRequestsPerSecond) o el número real de solicitudes. El rendimiento real de solicitudes utilizado se calcula como el promedio requests/second en un período de una hora. Recomendamos empezar por el valor predeterminado`minRecommendationRequestsPerSecond`, realizar un seguimiento del uso mediante CloudWatch las métricas de Amazon y, a continuación, aumentarlas `minRecommendationRequestsPerSecond` según sea necesario.

Estado

Un recomendador puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

Para obtener el estado de recomendante, llama a DescribeRecommender.

> **Note**
>
> Espere a que el valor `status` del recomendador sea `ACTIVE` antes de pedirle recomendaciones.

API relacionadas

- ListRecommenders
- DescribeRecommender
- UpdateRecommender
- DeleteRecommender

Sintaxis de la solicitud

```
{
   "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"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos del dominio destino para l recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

name

El nombre del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

recipeArn

El nombre de recurso de Amazon (ARN) de la receta que usará el recomendador. Para un recomendador, una receta es un caso de uso de un grupo de conjuntos de datos de dominio. Solo se pueden usar los casos de uso de grupos de conjuntos de datos de dominio para crear un recomendador. Para obtener información sobre los casos de uso, consulte Elección de los casos de uso de un recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

recommenderConfig

Los detalles de configuración del recomendador.

Tipo: objeto RecommenderConfig

Obligatorio: no

tags

Lista de etiquetas que se aplican al recomendador.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "recommenderArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateSchema

Servicio: Amazon Personalize

Crea un esquema de Amazon Personalize a partir de la cadena de esquema especificada. El esquema que cree debe estar en formato Avro JSON.

Amazon Personalize reconoce tres variantes de esquema. Cada esquema está asociado a un tipo de conjunto de datos y tiene un conjunto de campos y palabras clave necesarios. Si va a crear un esquema para un conjunto de datos en un grupo de conjunto de datos de dominio, debe proporcionar el dominio del grupo de conjunto de datos de dominio. Especificas un esquema cuando llamas CreateDataset.

Para obtener más información sobre los esquemas, consulte Datasets and schemas (Conjuntos de datos y esquemas).

API relacionadas

- ListSchemas
- DescribeSchema
- DeleteSchema

Sintaxis de la solicitud

```
{
   "domain": "string",
   "name": "string",
   "schema": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

domain

El dominio para el esquema. Si va a crear un esquema para un conjunto de datos en un grupo de conjuntos de datos de dominio, especifique el dominio que eligió al crear el grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

name

El nombre para el esquema.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

schema

Esquema en formato Avro JSON.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 20 000.

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "schemaArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

schemaArn

Nombre de recurso de Amazon (ARN) del esquema creado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateSolution

Servicio: Amazon Personalize

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Crea la configuración para entrenar un modelo (creación de una versión de solución). Esta configuración incluye la receta que se utilizará en el entrenamiento del modelo y la configuración de entrenamiento opcional, como las columnas que se utilizarán en el entrenamiento y los parámetros de transformación de características. Para obtener más información sobre la configuración de una solución, consulte Creación y configuración de una solución.

De forma predeterminada, las soluciones nuevas utilizan el entrenamiento automático para crear versiones de solución cada siete días. Puede cambiar la frecuencia de entrenamiento. La creación automática de versiones de solución comienza una hora después de que el estado de la solución sea ACTIVE. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático. Para obtener más información, consulte Configuración del entrenamiento automático.

Para desactivar el entrenamiento automático, establezca `performAutoTraining` en false. Si desactiva el entrenamiento automático, debe crear manualmente una versión de la solución llamando a la CreateSolutionVersion operación.

Cuando comience el entrenamiento, podrá obtener el nombre de recurso de Amazon (ARN) de la versión de la solución con la operación de la ListSolutionVersions API. Para obtener su estado, utilice el DescribeSolutionVersion.

Una vez finalizada la capacitación, puede evaluar la precisión del modelo llamando GetSolutionMetrics. Cuando esté satisfecho con la versión de la solución, la implementa usando CreateCampaign. La campaña proporciona recomendaciones a un cliente a través de la GetRecommendations API.

> **Note**
>
> Amazon Personalize no admite la configuración de `hpoObjective` para la optimización de hiperparámetros de solución en este momento.

Estado

Una solución puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Para conocer el estado de la solución, llame DescribeSolution. Si utiliza el entrenamiento manual, el estado debe ser ACTIVE antes de llamar a `CreateSolutionVersion`.

API relacionadas

- UpdateSolution
- ListSolutions
- CreateSolutionVersion
- DescribeSolution
- DeleteSolution

- ListSolutionVersions
- DescribeSolutionVersion

Sintaxis de la solicitud

```
{
   "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"
    }
  ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que proporciona los datos de entrenamiento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

eventType

Si tiene varios tipos de eventos (mediante un campo de esquema `EVENT_TYPE`), este parámetro especifica qué tipo de evento (por ejemplo, “clic” o “me gusta”) se utiliza para entrenar el modelo.

Si no se proporciona `eventType`, Amazon Personalize utiliza todas las interacciones para el entrenamiento con la misma ponderación, independientemente del tipo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

name

El nombre de la solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: sí

performAutoML

> **Important**
>
> No le recomendamos habilitar el machine learning automático. En lugar de eso, puede emparejar su caso de uso con las recetas de Amazon Personalize disponibles. Para obtener más información, consulte Elección de una receta.

Si se debe realizar un machine learning automatizado (AutoML). El valor predeterminado es `false`. En este caso, debe especificar `recipeArn`.

Cuando se configura en`true`, Amazon Personalize analiza los datos de entrenamiento y selecciona la receta y los hiperparámetros óptimos de USER_PERSONALIZATION. En este caso, debe omitir `recipeArn`. Amazon Personalize determina la receta óptima mediante la ejecución de pruebas con valores diferentes para los hiperparámetros. AutoML alarga el proceso de entrenamiento en comparación con la selección de una receta específica.

Tipo: Booleano

Obligatorio: no

performAutoTraining

Indica si la solución utiliza el entrenamiento automático para crear nuevas versiones de solución (modelos entrenados). El valor predeterminado es `True` y la solución crea automáticamente nuevas versiones de solución cada siete días. Puede cambiar la frecuencia de entrenamiento mediante la especificación de `schedulingExpression` en `AutoTrainingConfig` como parte de la configuración de la solución. Para obtener más información sobre el entrenamiento automático, consulte Configuración del entrenamiento automático.

La creación automática de versiones de solución comienza una hora después de que el estado de la solución sea ACTIVE. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático.

Cuando comience el entrenamiento, podrá obtener el nombre de recurso de Amazon (ARN) de la versión de la solución con la operación de la ListSolutionVersions API. Para obtener su estado, utilice el DescribeSolutionVersion.

Tipo: Booleano

Obligatorio: no

performHPO

Si se debe realizar la optimización de hiperparámetros (HPO) en la receta especificada o seleccionada. El valor predeterminado es `false`.

Al ejecutar AutoML, este parámetro es siempre `true` y no debe configurarlo en `false`.

Tipo: Booleano

Obligatorio: no

performIncrementalUpdate

Si desea realizar actualizaciones de entrenamiento incrementales en su modelo. Cuando está habilitada, esto permite que el modelo aprenda de los datos nuevos con mayor frecuencia sin necesidad de volver a capacitarlo por completo, lo que permite la personalización casi en tiempo real. Este parámetro solo es compatible con soluciones que utilizan la receta de similitud semántica.

Tipo: Booleano

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta que se va a utilizar para el entrenamiento del modelo. Esto es obligatorio cuando `performAutoML` es falso. Para obtener información sobre las distintas recetas de Amazon Personalize y sus ARN, consulte Elección de una receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionConfig

Propiedades de configuración de la solución. Si `performAutoML` se establece en true, Amazon Personalize solo evalúa la sección `autoMLConfig` de la configuración de la solución.

> **Note**
>
> Amazon Personalize no admite la configuración de `hpoObjective` en este momento.

Tipo: objeto SolutionConfig

Obligatorio: no

tags

Lista de etiquetas que se aplican a la solución.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "solutionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

solutionArn

El ARN de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

CreateSolutionVersion

Servicio: Amazon Personalize

Entrena o reentrena una solución activa en un grupo de conjuntos de datos personalizados. Se crea una solución mediante la CreateSolution operación y debe estar en estado ACTIVO antes de llamar`CreateSolutionVersion`. Se crea una nueva versión de la solución cada vez que llama a esta operación.

Estado

Una versión de la solución puede tener uno de los siguientes estados:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING
- CREATE STOPPED

Para obtener el estado de la versión, llame DescribeSolutionVersion. Espere a que el estado se muestre como ACTIVE antes de llamar a `CreateCampaign`.

Si el estado muestra CREATE FAILED, la respuesta incluye una clave `failureReason` que describe el motivo del error en el trabajo.

API relacionadas

- ListSolutionVersions
- DescribeSolutionVersion
- ListSolutions
- CreateSolution
- DescribeSolution
- DeleteSolution

Sintaxis de la solicitud

```
{
```

```
   "name": "string",
   "solutionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ],
   "trainingMode": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

name

El nombre de la versión de la solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

solutionArn

El nombre de recurso de Amazon (ARN) de la solución que contiene la información de configuración de entrenamiento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Lista de etiquetas que se aplican a la versión de solución.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: no

trainingMode

El alcance del entrenamiento que se realizará al crear la versión de la solución. El valor predeterminado es FULL. Esto crea un modelo completamente nuevo basado en la totalidad de los datos de entrenamiento de los conjuntos de datos de su grupo de conjuntos de datos.

Si la usas User-Personalization, puedes especificar un modo de entrenamiento deUPDATE. Esto actualiza el modelo para tener en cuenta nuevos elementos a modo de recomendaciones. No se trata de un reentrenamiento completo. Aún así, debe completar un reentrenamiento completo semanalmente. Si especifica UPDATE, Amazon Personalize detendrá las actualizaciones automáticas de la versión de la solución. Para reanudar las actualizaciones, cree una nueva solución con el modo de entrenamiento configurado en FULL e impleméntela en una campaña. Para obtener más información acerca de las actualizaciones automáticas, consulte Actualizaciones automáticas.

La UPDATE opción solo se puede usar cuando ya se ha creado una versión de solución activa a partir de la solución de entrada que utiliza la FULL opción y la solución de entrada se entrenó con la User-Personalization receta o la HRNN-Coldstart receta anterior.

Tipo: cadena

Valores válidos: FULL | UPDATE | AUTOTRAIN

Obligatorio: no

Sintaxis de la respuesta

```
{
   "solutionVersionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

solutionVersionArn

El ARN de la nueva versión de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteCampaign

Servicio: Amazon Personalize

Elimina una campaña al eliminar la implementación de la solución. La solución en la que se basa la campaña no se elimina y se puede volver a implementar cuando sea necesario. Ya no se puede especificar una campaña eliminada en una GetRecommendations solicitud. Para obtener información sobre la creación de campañas, consulte CreateCampaign.

Sintaxis de la solicitud

```
{
   "campaignArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteDataset

Servicio: Amazon Personalize

Elimina un conjunto de datos. No puede eliminar un conjunto de datos si un `DatasetImportJob` o `SolutionVersion` asociado se encuentra en el estado CREATE PENDING o IN PROGRESS. Para obtener más información sobre cómo eliminar conjuntos de datos, consulte Eliminación de un conjunto de datos.

Sintaxis de la solicitud

```
{
   "datasetArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteDatasetGroup

Servicio: Amazon Personalize

Elimina un grupo de conjuntos de datos. Antes de eliminar un grupo de conjuntos de datos, debe eliminar lo siguiente:

- Todos los rastreadores de eventos asociados.
- Todas las soluciones asociadas.
- Todos los conjuntos de datos del grupo de conjuntos de datos.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El ARN del grupo de conjuntos de datos que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteEventTracker

Servicio: Amazon Personalize

Elimina el rastreador de eventos. No elimina el conjunto de datos del grupo de conjuntos de datos. Para obtener más información sobre los rastreadores de eventos, consulte CreateEventTracker.

Sintaxis de la solicitud

```
{
   "eventTrackerArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

eventTrackerArn

El nombre de recurso de Amazon (ARN) del rastreador de eventos que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteFilter

Servicio: Amazon Personalize

Elimina un filtro.

Sintaxis de la solicitud

```
{
   "filterArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

filterArn

El ARN del filtro que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteMetricAttribution

Servicio: Amazon Personalize

Elimina una atribución de métricas.

Sintaxis de la solicitud

```
{
   "metricAttributionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

metricAttributionArn

El nombre de recurso de Amazon (ARN) de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteRecommender

Servicio: Amazon Personalize

Desactiva y elimina un recomendador. Ya no se puede especificar un recomendador eliminado en una GetRecommendations solicitud.

Sintaxis de la solicitud

```
{
   "recommenderArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteSchema

Servicio: Amazon Personalize

Elimina un esquema. Antes de eliminar un esquema, debe eliminar todos los conjuntos de datos que hacen referencia al esquema. Para obtener más información sobre los esquemas, consulte CreateSchema.

Sintaxis de la solicitud

```
{
   "schemaArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

schemaArn

Nombre de recurso de Amazon (ARN) del esquema que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DeleteSolution

Servicio: Amazon Personalize

Elimina todas las versiones de una solución y el objeto `Solution` en sí. Antes de eliminar una solución, debe eliminar todas las campañas basadas en la solución. Para determinar qué campañas utilizan la solución, llama ListCampaigns y proporciona el nombre de recurso de Amazon (ARN) de la solución. No puede eliminar una solución si un `SolutionVersion` asociado se encuentra en el estado CREATE PENDING o IN PROGRESS. Para obtener más información sobre las soluciones, consulte CreateSolution.

Sintaxis de la solicitud

```
{
   "solutionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

solutionArn

El ARN de la solución que se va a eliminar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeAlgorithm

Servicio: Amazon Personalize

Describe el algoritmo dado.

Sintaxis de la solicitud

```
{
   "algorithmArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

algorithmArn

El nombre de recurso de Amazon (ARN) del algoritmo que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

algorithm

Una lista de las propiedades del algoritmo.

Tipo: objeto Algorithm

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeBatchInferenceJob

Servicio: Amazon Personalize

Obtiene las propiedades de un trabajo de inferencia por lotes, incluidos el nombre, el nombre del recurso de Amazon (ARN), el estado, las configuraciones de entrada y salida y el ARN de la versión de la solución utilizada para generar las recomendaciones.

Sintaxis de la solicitud

```
{
   "batchInferenceJobArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

batchInferenceJobArn

El ARN del trabajo de inferencia por lotes que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
        }
      }
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchInferenceJob

Información sobre el trabajo de inferencia por lotes especificado.

Tipo: objeto BatchInferenceJob

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeBatchSegmentJob

Servicio: Amazon Personalize

Obtiene las propiedades de un trabajo de segmentos por lotes, incluidos el nombre, el nombre del recurso de Amazon (ARN), el estado, las configuraciones de entrada y salida y el ARN de la versión de la solución utilizada para generar segmentos.

Sintaxis de la solicitud

```
{
   "batchSegmentJobArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

batchSegmentJobArn

El ARN del trabajo de segmento por lotes que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchSegmentJob

Información sobre el trabajo de segmentos por lotes especificado.

Tipo: objeto BatchSegmentJob

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeCampaign

Servicio: Amazon Personalize

Describe el la campaña indicada, incluido su estado.

Una campaña puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Cuando `status` es `CREATE FAILED`, la respuesta incluye la clave `failureReason`, que describe el motivo.

Para obtener más información sobre las campañas, consulte CreateCampaign.

Sintaxis de la solicitud

```
{
   "campaignArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

campaign

> **Note**
>
> El campo `latestCampaignUpdate` solo se devuelve cuando la campaña ha tenido al menos una llamada de `UpdateCampaign`.

Las propiedades de la campaña.

> **Note**
>
> El campo `latestCampaignUpdate` solo se devuelve cuando la campaña ha tenido al menos una llamada de `UpdateCampaign`.

Tipo: objeto Campaign

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++

- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeDataDeletionJob

Servicio: Amazon Personalize

Describe el trabajo de eliminación de datos creado por CreateDataDeletionJob, incluido el estado del trabajo.

Sintaxis de la solicitud

```
{
   "dataDeletionJobArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

dataDeletionJobArn

Nombre de recurso de Amazon (ARN) del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "dataDeletionJob": {
      "creationDateTime": number,
      "dataDeletionJobArn": "string",
      "datasetGroupArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "numDeleted": number,
```

```
        "roleArn": "string",
        "status": "string"
    }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

dataDeletionJob

Información sobre el trabajo de eliminación de datos, incluido el estado.

El estado es uno de los siguientes valores:

- PENDIENTE
- IN_PROGRESS
- COMPLETED
- ERROR

Tipo: objeto DataDeletionJob

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeDataset

Servicio: Amazon Personalize

Describe el conjunto de datos especificado. Para obtener más información sobre los conjuntos de datos, consulte CreateDataset.

Sintaxis de la solicitud

```
{
   "datasetArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
    }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

dataset

Una lista de las propiedades del conjunto de datos.

Tipo: objeto Dataset

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4

- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeDatasetExportJob

Servicio: Amazon Personalize

Describe el trabajo de exportación del conjunto de datos creado por CreateDatasetExportJob, incluido el estado del trabajo de exportación.

Sintaxis de la solicitud

```
{
   "datasetExportJobArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetExportJobArn

El nombre de recurso de Amazon (ARN) del trabajo de exportación de conjuntos de datos que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetExportJob

Información sobre el trabajo de exportación de conjuntos de datos, incluido el estado.

El estado es uno de los siguientes valores:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: objeto DatasetExportJob

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeDatasetGroup

Servicio: Amazon Personalize

Describe el grupo de conjuntos de datos especificado. Para obtener más información sobre los grupos de conjuntos de datos, consulte CreateDatasetGroup.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "datasetGroup": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "domain": "string",
      "failureReason": "string",
      "kmsKeyArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "roleArn": "string",
      "status": "string"
   }
```

```
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetGroup

Una lista de las propiedades del grupo de conjuntos de datos.

Tipo: objeto DatasetGroup

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin

- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeDatasetImportJob

Servicio: Amazon Personalize

Describe el trabajo de importación del conjunto de datos creado por CreateDatasetImportJob, incluido el estado del trabajo de importación.

Sintaxis de la solicitud

```
{
   "datasetImportJobArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetImportJobArn

El nombre de recurso de Amazon (ARN) de la tarea de importación del conjunto de datos que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "datasetImportJob": {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetImportJobArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "importMode": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
```

```
      "publishAttributionMetricsToS3": boolean,
      "roleArn": "string",
      "status": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetImportJob

Información sobre el trabajo de importación del conjunto de datos, incluido el estado.

El estado es uno de los siguientes valores:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: objeto DatasetImportJob

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeEventTracker

Servicio: Amazon Personalize

Describe un rastreador de eventos. La respuesta incluye el valor `trackingId` y `status` del rastreador de eventos. Para obtener más información sobre los rastreadores de eventos, consulte CreateEventTracker.

Sintaxis de la solicitud

```
{
   "eventTrackerArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

eventTrackerArn

El nombre de recurso de Amazon (ARN) del rastreador de eventos que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "eventTracker": {
      "accountId": "string",
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "eventTrackerArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string",
      "trackingId": "string"
   }
```

```
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

eventTracker

Un objeto que describe el rastreador de eventos.

Tipo: objeto EventTracker

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin

- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeFeatureTransformation

Servicio: Amazon Personalize

Describe la transformación de la característica dada.

Sintaxis de la solicitud

```
{
   "featureTransformationArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

featureTransformationArn

El nombre de recurso de Amazon (ARN) de la transformación de característica que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "featureTransformation": {
      "creationDateTime": number,
      "defaultParameters": {
         "string" : "string"
      },
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
```

```
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

featureTransformation

Una lista de las FeatureTransformation propiedades.

Tipo: objeto FeatureTransformation

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin

- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeFilter

Servicio: Amazon Personalize

Describe las propiedades del filtro.

Sintaxis de la solicitud

```
{
   "filterArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

filterArn

El ARN del filtro que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "filter": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "filterArn": "string",
      "filterExpression": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

filter

Los detalles del filtro.

Tipo: objeto Filter

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3

- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeMetricAttribution

Servicio: Amazon Personalize

Describe una atribución de métricas.

Sintaxis de la solicitud

```
{
   "metricAttributionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

metricAttributionArn

El nombre de recurso de Amazon (ARN) de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "metricAttribution": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "metricAttributionArn": "string",
      "metricsOutputConfig": {
         "roleArn": "string",
         "s3DataDestination": {
            "kmsKeyArn": "string",
            "path": "string"
         }
```

```
        },
        "name": "string",
        "status": "string"
    }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metricAttribution

Los detalles de la atribución de métricas.

Tipo: objeto MetricAttribution

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2

- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeRecipe

Servicio: Amazon Personalize

Describe una receta.

Una receta contiene tres elementos:

- Un algoritmo que entrena un modelo.
- Hiperparámetros que rigen el entrenamiento.
- Información de transformación de características para modificar los datos de entrada antes del entrenamiento.

Amazon Personalize proporciona un conjunto de recetas predefinidas. Usted especifica una receta cuando crea una solución con la CreateSolution API. `CreateSolution`entrena un modelo mediante el algoritmo de la receta especificada y un conjunto de datos de entrenamiento. La solución, cuando se implementa como una campaña, puede ofrecer recomendaciones mediante la GetRecommendations API.

Sintaxis de la solicitud

```
{
   "recipeArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recipeArn

El nombre de recurso de Amazon (ARN) de la receta que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "recipe": {
      "algorithmArn": "string",
      "creationDateTime": number,
      "description": "string",
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "recipeType": "string",
      "status": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recipe

Un objeto que describe la receta.

Tipo: objeto Recipe

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos para cada idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeRecommender

Servicio: Amazon Personalize

Describe el recomendador indicado, incluido su estado.

Un recomendador puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

Cuando `status` es `CREATE FAILED`, la respuesta incluye la clave `failureReason`, que describe el motivo.

La clave `modelMetrics` es nula cuando se crea o se elimina el recomendador.

Para obtener más información sobre los recomendadores, consulte CreateRecommender.

Sintaxis de la solicitud

```
{
   "recommenderArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recommender

Las propiedades del recomendador.

Tipo: objeto Recommender

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2

- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeSchema

Servicio: Amazon Personalize

Describe un esquema. Para obtener más información sobre los esquemas, consulte CreateSchema.

Sintaxis de la solicitud

```
{
   "schemaArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

schemaArn

Nombre de recurso de Amazon (ARN) del esquema que se va a recuperar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "schema": {
      "creationDateTime": number,
      "domain": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "schema": "string",
      "schemaArn": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

schema

El esquema solicitado.

Tipo: objeto DatasetSchema

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeSolution

Servicio: Amazon Personalize

Describe una solución. Para obtener más información sobre las soluciones, consulte CreateSolution.

Sintaxis de la solicitud

```
{
   "solutionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

solutionArn

El nombre de recurso de Amazon (ARN) de la solución que se va a describir.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

solution

Un objeto que describe la solución.

Tipo: objeto Solution

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

DescribeSolutionVersion

Servicio: Amazon Personalize

Describe una versión específica de una solución. Para obtener más información sobre las soluciones, consulte CreateSolution

Sintaxis de la solicitud

```
{
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "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"
        }
    }
  }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

solutionVersion

La versión de la solución.

Tipo: objeto SolutionVersion

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

GetSolutionMetrics

Servicio: Amazon Personalize

Obtiene las métricas de la versión de solución especificada.

Sintaxis de la solicitud

```
{
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución para la que se van a obtener métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "metrics": {
      "string" : number
   },
   "solutionVersionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metrics

Las métricas para la versión de la solución. Para obtener más información, consulte Evaluación de una versión de solución con métricas.

Tipo: cadena al mapa doble

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

solutionVersionArn

El mismo ARN de la versión de solución especificado en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListBatchInferenceJobs

Servicio: Amazon Personalize

Obtiene una lista de los trabajos de inferencia por lotes que se han realizado a partir de una versión de solución.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

El número máximo de resultados del trabajo de inferencia por lotes que se devolverán en cada página. El valor predeterminado es 100.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

El token para solicitar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución a partir de la que se crearon trabajos de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "batchInferenceJobs": [
      {
         "batchInferenceJobArn": "string",
         "batchInferenceJobMode": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchInferenceJobs

Una lista que contiene información sobre cada trabajo que se devuelve.

Tipo: matriz de objetos BatchInferenceJobSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

El token que se usará para recuperar la siguiente página de resultados. El valor es `null` cuando no hay más resultados que devolver.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListBatchSegmentJobs

Servicio: Amazon Personalize

Obtiene una lista de los trabajos de segmento por lotes que se han realizado a partir de una versión de solución que especifique.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

El número máximo de resultados del trabajo de segmento por lotes que se devolverán en cada página. El valor predeterminado es 100.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

El token para solicitar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución que los trabajos de segmento por lotes utilizaron para generar segmentos por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "batchSegmentJobs": [
      {
         "batchSegmentJobArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

batchSegmentJobs

Una lista que contiene información sobre cada trabajo que se devuelve.

Tipo: matriz de objetos BatchSegmentJobSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

El token que se usará para recuperar la siguiente página de resultados. El valor es `null` cuando no hay más resultados que devolver.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListCampaigns

Servicio: Amazon Personalize

Devuelve una lista de campañas que utilizan la solución dada. Cuando no se especifica una solución, se muestran todas las campañas asociadas a la cuenta. La respuesta proporciona las propiedades para cada campaña, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre las campañas, consulte CreateCampaign.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

Número máximo de filas que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto en la convocatoria anterior ListCampaigns para obtener el siguiente conjunto de campañas (si existe).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

solutionArn

El nombre de recurso de Amazon (ARN) de la solución para la que se mostrarán las campañas. Cuando no se especifica una solución, se muestran todas las campañas asociadas a la cuenta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "campaigns": [
      {
         "campaignArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

campaigns

Una lista de las campañas.

Tipo: matriz de objetos CampaignSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de campañas (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListDataDeletionJobs

Servicio: Amazon Personalize

Devuelve una lista de los trabajos de eliminación de datos de un grupo de conjuntos de datos ordenados por hora de creación, con el más reciente en primer lugar. Cuando no se especifica un grupo de conjuntos de datos, se enumeran todos los trabajos de eliminación de datos asociados a la cuenta. La respuesta proporciona las propiedades para cada trabajo, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los trabajos de eliminación de datos, consulte Eliminación de usuarios.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

Nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos del que se enumeran los trabajos de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de trabajos de eliminación de datos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListDataDeletionJobs` para obtener el siguiente conjunto de trabajos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "dataDeletionJobs": [
      {
         "creationDateTime": number,
         "dataDeletionJobArn": "string",
         "datasetGroupArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

dataDeletionJobs

Lista de trabajos de eliminación de datos.

Tipo: matriz de objetos DataDeletionJobSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de trabajos de eliminación de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)

- AWS SDK para Ruby V3

ListDatasetExportJobs

Servicio: Amazon Personalize

Devuelve una lista de trabajos de exportación de conjuntos de datos que utilizan el conjunto de datos dado. Cuando no se especifica un conjunto de datos, se muestran todos los trabajos de exportación de conjuntos de datos asociados a la cuenta. La respuesta proporciona las propiedades para cada trabajo de exportación de conjuntos de datos, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los trabajos de exportación de conjuntos de datos, consulte CreateDatasetExportJob. Para obtener más información sobre los conjuntos de datos, consulte CreateDataset.

Sintaxis de la solicitud

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos para el que se enumeran los trabajos de exportación de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de trabajos de exportación de conjuntos de datos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListDatasetExportJobs` para obtener el siguiente conjunto de trabajos de exportación de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetExportJobs": [
      {
         "creationDateTime": number,
         "datasetExportJobArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetExportJobs

Lista de trabajos de exportación de conjuntos de datos.

Tipo: matriz de objetos DatasetExportJobSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de trabajos de exportación de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3

- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListDatasetGroups

Servicio: Amazon Personalize

Devuelve una lista de grupos de conjuntos de datos. La respuesta proporciona las propiedades para cada grupo de conjuntos de datos, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los grupos de conjuntos de datos, consulte CreateDatasetGroup.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

Número máximo de grupos de conjuntos de datos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListDatasetGroups` para obtener el siguiente conjunto de grupos de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
```

```
   "datasetGroups": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "domain": "string",
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetGroups

La lista de sus grupos de conjuntos de datos.

Tipo: matriz de objetos DatasetGroupSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de grupos de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListDatasetImportJobs

Servicio: Amazon Personalize

Devuelve una lista de trabajos de importación de conjuntos de datos que utilizan el conjunto de datos dado. Cuando no se especifica un conjunto de datos, se muestran todos los trabajos de importación de conjuntos de datos asociados a la cuenta. La respuesta proporciona las propiedades para cada trabajo de importación de conjuntos de datos, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los trabajos de importación de conjuntos de datos, consulte CreateDatasetImportJob. Para obtener más información sobre los conjuntos de datos, consulte CreateDataset.

Sintaxis de la solicitud

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos para el que se enumeran los trabajos de importación de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de trabajos de importación de conjuntos de datos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListDatasetImportJobs` para obtener el siguiente conjunto de trabajos de importación de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasetImportJobs": [
      {
         "creationDateTime": number,
         "datasetImportJobArn": "string",
         "failureReason": "string",
         "importMode": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetImportJobs

Lista de trabajos de importación de conjuntos de datos.

Tipo: matriz de objetos DatasetImportJobSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de trabajos de importación de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3

- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListDatasets

Servicio: Amazon Personalize

Devuelve la lista de conjuntos de datos contenidos en el grupo de conjuntos de datos dado. La respuesta proporciona las propiedades para cada conjunto de datos, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los conjuntos de datos, consulte CreateDataset.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que contiene los conjuntos de datos que se van a enumerar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de conjuntos de datos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListDatasets` para obtener el siguiente conjunto de trabajos de importación de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "datasets": [
      {
         "creationDateTime": number,
         "datasetArn": "string",
         "datasetType": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasets

Una matriz de objetos `Dataset`. Cada objeto proporciona información de metadatos.

Tipo: matriz de objetos DatasetSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de conjuntos de datos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListEventTrackers

Servicio: Amazon Personalize

Devuelve la lista de rastreadores de eventos asociados a la cuenta. La respuesta proporciona las propiedades para cada rastreador de eventos, incluido el nombre de recurso de Amazon (ARN) y el ID de seguimiento. Para obtener más información sobre los rastreadores de eventos, consulte CreateEventTracker.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El ARN de un grupo de conjuntos de datos utilizado para filtrar la respuesta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de rastreadores de eventos que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListEventTrackers` para obtener el siguiente conjunto de rastreadores de eventos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "eventTrackers": [
      {
         "creationDateTime": number,
         "eventTrackerArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

eventTrackers

Una lista de rastreadores de eventos.

Tipo: matriz de objetos EventTrackerSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de rastreadores de eventos (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListFilters

Servicio: Amazon Personalize

Enumera todos los filtros que pertenecen a un grupo de conjuntos de datos determinado.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El ARN del grupo de conjuntos de datos que contiene los filtros.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de filtros que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListFilters` para obtener el siguiente conjunto de filtros (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "Filters": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "failureReason": "string",
         "filterArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Filters

Lista de filtros devueltos.

Tipo: matriz de objetos FilterSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Un token para obtener el siguiente conjunto de filtros (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListMetricAttributionMetrics

Servicio: Amazon Personalize

Enumera las métricas de la atribución de métricas.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "metricAttributionArn": "string",
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

El número máximo de métricas que se devolverán en una página de resultados.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

metricAttributionArn

El nombre de recurso de Amazon (ARN) de la atribución de métricas para la que se recuperarán atributos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

nextToken

Especifique el token de paginación de una solicitud anterior para recuperar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metrics

Las métricas de la atribución de métricas especificada.

Tipo: matriz de objetos MetricAttribute

Miembros de la matriz: número máximo de 10 elementos.

nextToken

Especifique el token de paginación de una solicitud `ListMetricAttributionMetricsResponse` anterior para recuperar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListMetricAttributions

Servicio: Amazon Personalize

Enumera las atribuciones de métricas.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de las atribuciones de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

El número máximo de atribuciones de métricas que se devolverán en una página de resultados.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Especifique el token de paginación de una solicitud anterior para recuperar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "metricAttributions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "metricAttributionArn": "string",
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metricAttributions

La lista de atribuciones de métricas.

Tipo: matriz de objetos MetricAttributionSummary

Miembros de la matriz: número máximo de 100 elementos.

nextToken

Especifique el token de paginación de una solicitud anterior para recuperar la siguiente página de resultados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListRecipes

Servicio: Amazon Personalize

Devuelve una lista de recetas disponibles. La respuesta proporciona las propiedades para cada receta, incluido el nombre de recurso de Amazon (ARN) de la receta.

Sintaxis de la solicitud

```
{
   "domain": "string",
   "maxResults": number,
   "nextToken": "string",
   "recipeProvider": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

domain

Filtra recetas devueltas por dominio para un grupo de conjuntos de datos de dominio. En la respuesta solo se incluyen las recetas (casos de uso del grupo de conjuntos de datos de dominio) para este dominio. Si no especifica ningún dominio, se devuelven todas las recetas.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

maxResults

El número máximo de recetas que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListRecipes` para obtener el siguiente conjunto de recetas (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

recipeProvider

El valor predeterminado es `SERVICE`.

Tipo: cadena

Valores válidos: `SERVICE`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "nextToken": "string",
   "recipes": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "status": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

nextToken

Un token para obtener el siguiente conjunto de recetas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

recipes

La lista de recetas disponibles.

Tipo: matriz de objetos RecipeSummary

Miembros de la matriz: número máximo de 100 elementos.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin

- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListRecommenders

Servicio: Amazon Personalize

Devuelve una lista de los recomendadores de un grupo de conjuntos de datos de dominio determinado. Cuando no se especifica un grupo de conjuntos de datos de dominio, se muestran todos los recomendadores asociados a la cuenta. La respuesta proporciona las propiedades de cada recomendador, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los recomendadores, consulte CreateRecommender.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de dominio para el que se van a enumerar los recomendadores. Cuando no se especifica un grupo de conjuntos de datos de dominio, se muestran todos los recomendadores asociados a la cuenta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

El número máximo de recomendadores que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListRecommenders` para obtener el siguiente conjunto de recomendadores (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "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"
      }
   ]
```

```
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

nextToken

Un token para obtener el siguiente conjunto de recomendaciones (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

recommenders

Una lista de los recomendadores.

Tipo: matriz de objetos RecommenderSummary

Miembros de la matriz: número máximo de 100 elementos.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListSchemas

Servicio: Amazon Personalize

Devuelve la lista de esquemas asociados a la cuenta. La respuesta proporciona las propiedades para cada esquema, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre los esquemas, consulte CreateSchema.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

Número máximo de esquemas que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListSchemas` para obtener el siguiente conjunto de esquemas (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
```

```
   "nextToken": "string",
   "schemas": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "schemaArn": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

nextToken

Un token que se utiliza para obtener el siguiente conjunto de esquemas (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

schemas

Una lista de esquemas.

Tipo: matriz de objetos DatasetSchemaSummary

Miembros de la matriz: número máximo de 100 elementos.

Errores

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListSolutions

Servicio: Amazon Personalize

Devuelve una lista de soluciones de un grupo de conjuntos de datos dado. Cuando no se especifica un grupo de conjuntos de datos, se muestran todas las soluciones asociadas a la cuenta. La respuesta proporciona las propiedades para cada solución, incluido el nombre de recurso de Amazon (ARN). Para obtener más información sobre las soluciones, consulte CreateSolution.

Sintaxis de la solicitud

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

maxResults

Número máximo de soluciones que se devolverán.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListSolutions` para obtener el siguiente conjunto de soluciones (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "nextToken": "string",
   "solutions": [
      {
         "creationDateTime": number,
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "solutionArn": "string",
         "status": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

nextToken

Un token para obtener el siguiente conjunto de soluciones (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

solutions

Una lista de las soluciones actuales.

Tipo: matriz de objetos SolutionSummary

Miembros de la matriz: número máximo de 100 elementos.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulte lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListSolutionVersions

Servicio: Amazon Personalize

Devuelve una lista de las versiones de solución de la solución dada. Cuando no se especifica una solución, se muestran todas las versiones de solución asociadas a la cuenta. La respuesta proporciona las propiedades para cada versión de solución, incluido el nombre de recurso de Amazon (ARN).

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

maxResults

Número máximo de versiones de solución que se van a devolver.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

nextToken

Un token devuelto por la llamada anterior a `ListSolutionVersions` para obtener el siguiente conjunto de versiones de solución (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

Obligatorio: no

solutionArn

El nombre de recurso de Amazon (ARN) de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "nextToken": "string",
   "solutionVersions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string",
         "trainingMode": "string",
         "trainingType": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

nextToken

Un token para obtener el siguiente conjunto de versiones de solución (si existen).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1500 caracteres.

Patrón: `\p{ASCII}{0,1500}`

solutionVersions

Una lista de las versiones de solución que describe las propiedades de la versión.

Tipo: matriz de objetos SolutionVersionSummary

Miembros de la matriz: número máximo de 100 elementos.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

InvalidNextTokenException

El token no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

ListTagsForResource

Servicio: Amazon Personalize

Obtenga una lista de etiquetas adjuntas a un recurso.

Sintaxis de la solicitud

```
{
   "resourceArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

resourceArn

El nombre de recurso de Amazon (ARN) del recurso.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

tags

Etiquetas del recurso.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

StartRecommender

Servicio: Amazon Personalize

Inicia un recomendador con el estado INACTIVE. El inicio de un recomendador no crea ningún modelo nuevo, sino que reanuda la facturación y el reentrenamiento automático del recomendador.

Sintaxis de la solicitud

```
{
   "recommenderArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a iniciar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "recommenderArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador iniciado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)

- AWS SDK para Ruby V3

StopRecommender

Servicio: Amazon Personalize

Detiene un recomendador que está en estado ACTIVE. La detención de un recomendador detiene la facturación y su reentrenamiento automático.

Sintaxis de la solicitud

```
{
   "recommenderArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a detener.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "recommenderArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador detenido.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)

- AWS SDK para Ruby V3

StopSolutionVersionCreation

Servicio: Amazon Personalize

Detiene la creación de una versión de la solución que se encuentra en el estado CREATE_PENDING o CREATE IN_PROGRESS.

Según el estado actual de la versión de la solución, el estado de la versión de la solución cambia de la siguiente manera:

- CREATE_PENDING > CREATE_STOPPED

  o

- CREATE_IN_PROGRESS > CREATE_STOPPING > CREATE_STOPPED

Se le facturará todo el entrenamiento completado hasta que detenga la creación de la versión de la solución. No podrá reanudar la creación de una versión de la solución una vez que se haya detenido.

Sintaxis de la solicitud

```
{
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de la solución que desea dejar de crear.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

TagResource

Servicio: Amazon Personalize

Agregue una lista de etiquetas a un recurso.

Sintaxis de la solicitud

```
{
   "resourceArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

resourceArn

El nombre de recurso de Amazon (ARN) del recurso.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tags

Las etiquetas que se aplicarán al recurso. Para obtener más información, consulte Etiquetado de recursos de Amazon Personalize.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagsException

Ha superado el número máximo de etiquetas que puede aplicar a este recurso.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4

- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UntagResource

Servicio: Amazon Personalize

Elimina las etiquetas especificadas que están adjuntas a un recurso. Para obtener más información, consulte Eliminación de etiquetas de los recursos de Amazon Personalize.

Sintaxis de la solicitud

```
{
   "resourceArn": "string",
   "tagKeys": [ "string" ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

resourceArn

El nombre de recurso de Amazon (ARN) del recurso.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

tagKeys

Las claves de las etiquetas que se van a eliminar.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 0 artículos. La cantidad máxima es de 200 artículos.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

TooManyTagKeysException

La solicitud contiene más claves de etiqueta de las que se pueden asociar a un recurso (50 claves de etiqueta por recurso).

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2

- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UpdateCampaign

Servicio: Amazon Personalize

Actualiza una campaña para implementar una versión de solución reentrenada con una campaña existente, cambiar `minProvisionedTPS` de la campaña o modificar la configuración de la campaña. Por ejemplo, puede establecer `enableMetadataWithRecommendations` en true en el caso de una campaña existente.

Para actualizar una campaña a fin de que comience automáticamente con la versión más reciente de la solución, especifique lo siguiente:

- Para el parámetro `SolutionVersionArn`, especifique el nombre de recurso de Amazon (ARN) de la solución en el formato `SolutionArn/$LATEST`.
- En `campaignConfig`, establezca `syncWithLatestSolutionVersion` en `true`.

Para actualizar una campaña, el estado de la campaña debe ser ACTIVE o CREATE FAILED. Comprueba el estado de la campaña mediante la DescribeCampaign operación.

> **Note**
>
> Puede seguir recibiendo recomendaciones de una campaña mientras haya una actualización en curso. La campaña utilizará la versión de la solución y la configuración de la campaña anterior para generar recomendaciones hasta que el estado de la última actualización de la campaña sea `Active`.

Para obtener más información sobre la actualización de una campaña, incluidos ejemplos de código, consulte Actualización de una campaña. Para obtener información acerca de las campañas, consulte Creación de una campaña.

Sintaxis de la solicitud

```
{
   "campaignArn": "string",
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
```

```
      "rankingInfluence": {
         "string" : number
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
   "solutionVersionArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

campaignConfig

Los detalles de la configuración de una campaña.

Tipo: objeto CampaignConfig

Obligatorio: no

minProvisionedTPS

Especifica el número mínimo de transacciones aprovisionadas (recomendaciones) por segundo solicitadas que Amazon Personalize va a admitir. Un valor elevado de `minProvisionedTPS` aumentará su factura. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realiza un seguimiento de tu uso mediante CloudWatch las métricas de Amazon y `minProvisionedTPS` auméntalas según sea necesario.

Tipo: entero

Rango válido: valor mínimo de 1.

Obligatorio: no

solutionVersionArn

Nombre de recurso de Amazon (ARN) del nuevo modelo que se va a implementar. Para especificar la versión más reciente de la solución, especifique el ARN de la solución en formato `SolutionArn/$LATEST`. Debes usar este formato si lo `syncWithLatestSolutionVersion` configuraste `True` en CampaignConfig.

Para implementar un modelo que no sea la última versión de solución, especifique el ARN de la versión de solución.

Para obtener más información sobre las actualizaciones de campaña automáticas, consulte Habilitación de las actualizaciones de campaña automáticas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "campaignArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

campaignArn

El mismo ARN de campaña que figura en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UpdateDataset

Servicio: Amazon Personalize

Actualice un conjunto de datos para reemplazar su esquema por uno nuevo o existente. Para obtener más información, consulte Sustitución del esquema de un conjunto de datos.

Sintaxis de la solicitud

```
{
   "datasetArn": "string",
   "schemaArn": "string"
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que desea actualizar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

schemaArn

Nombre de recurso de Amazon (ARN) del nuevo esquema que desea utilizar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
{
```

```
    "datasetArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que ha actualizado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UpdateMetricAttribution

Servicio: Amazon Personalize

Actualiza una atribución de métricas.

Sintaxis de la solicitud

```
{
   "addMetrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricAttributionArn": "string",
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "removeMetrics": [ "string" ]
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

addMetrics

Agrega nuevos atributos de métrica a la atribución de métricas.

Tipo: matriz de objetos MetricAttribute

Miembros de la matriz: número máximo de 10 elementos.

Obligatorio: no

metricAttributionArn

El Nombre de recurso de Amazon (ARN) para la atribución de métricas que se va a actualizar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

metricsOutputConfig

Una configuración de salida para la atribución de métricas.

Tipo: objeto MetricAttributionOutput

Obligatorio: no

removeMetrics

Elimine los atributos de métricas de la atribución de métricas.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 10 elementos.

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Sintaxis de la respuesta

```
{
   "metricAttributionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

metricAttributionArn

El nombre de recurso de Amazon (ARN) para la atribución de métricas que ha actualizado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceAlreadyExistsException

El recurso especificado ya existe.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2

- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UpdateRecommender

Servicio: Amazon Personalize

Actualiza el recomendador para modificar su configuración. Si actualiza el recomendador para modificar las columnas utilizadas en el entrenamiento, Amazon Personalize inicia automáticamente un reentrenamiento completo de los modelos que respaldan al recomendador. Mientras se completa la actualización, podrá seguir recibiendo recomendaciones del recomendador. El recomendador utiliza la configuración anterior hasta que se complete la actualización. Para realizar un seguimiento del estado de esta actualización, utilice el `latestRecommenderUpdate` devuelto en la operación DescribeRecommender.

Sintaxis de la solicitud

```
{
   "recommenderArn": "string",
   "recommenderConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "minRecommendationRequestsPerSecond": number,
      "trainingDataConfig": {
         "excludedDatasetColumns": {
            "string" : [ "string" ]
         },
         "includedDatasetColumns": {
            "string" : [ "string" ]
         }
      }
   }
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a modificar.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

recommenderConfig

Los detalles de configuración del recomendador.

Tipo: objeto RecommenderConfig

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "recommenderArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

recommenderArn

El mismo nombre de recurso de Amazon (ARN) de recomendador que se indica en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

UpdateSolution

Servicio: Amazon Personalize

Actualiza una solución de Amazon Personalize para usar una configuración de entrenamiento automático diferente. Al actualizar una solución, puede cambiar si la solución utiliza el entrenamiento automático, así como la frecuencia del entrenamiento. Para obtener más información sobre la actualización de una solución, consulte Actualización de una solución.

Una actualización de solución puede tener uno de los siguientes estados:

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Para obtener el estado de una actualización de la solución, llame a la operación de la DescribeSolution API y busque el estado en `latestSolutionUpdate`.

Sintaxis de la solicitud

```
{
   "performAutoTraining": boolean,
   "performIncrementalUpdate": boolean,
   "solutionArn": "string",
   "solutionUpdateConfig": {
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
      "eventsConfig": {
         "eventParametersList": [
            {
               "eventType": "string",
               "eventValueThreshold": number,
               "weight": number
            }
         ]
      }
   }
}
```

Parámetros de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

performAutoTraining

Indica si la solución utiliza el entrenamiento automático para crear nuevas versiones de solución (modelos entrenados). Puede cambiar la frecuencia de entrenamiento mediante la especificación de `schedulingExpression` en `AutoTrainingConfig` como parte de la configuración de la solución.

Si activa el entrenamiento automático, el primero comenzará una hora después de que se complete la actualización de la solución. Si crea de forma manual una versión de solución en menos de una hora, la solución omitirá el primer entrenamiento automático. Para obtener más información sobre el entrenamiento automático, consulte Configuración del entrenamiento automático.

Cuando comience la capacitación, podrá obtener el nombre de recurso de Amazon (ARN) de la versión de la solución con la operación de la ListSolutionVersions API. Para obtener su estado, utilice el DescribeSolutionVersion.

Tipo: Booleano

Obligatorio: no

performIncrementalUpdate

Si desea realizar actualizaciones de entrenamiento incrementales en su modelo. Cuando está habilitada, esto permite que el modelo aprenda de los datos nuevos con mayor frecuencia sin necesidad de volver a capacitarlo por completo, lo que permite la personalización casi en tiempo real. Este parámetro solo es compatible con soluciones que utilizan la receta de similitud semántica.

Tipo: Booleano

Obligatorio: no

solutionArn

Nombre de recurso de Amazon (ARN) de la solución que se actualizará.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

solutionUpdateConfig

Nuevos detalles de configuración de la solución.

Tipo: objeto SolutionUpdateConfig

Obligatorio: no

Sintaxis de la respuesta

```
{
   "solutionArn": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

solutionArn

El mismo nombre de recurso de Amazon (ARN) de solución que se indica en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

LimitExceededException

Se ha excedido el límite en el número de solicitudes por segundo.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 400

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

Eventos de Amazon Personalize

Eventos de Amazon Personalize admite las siguientes acciones:

- PutActionInteractions
- PutActions
- PutEvents
- PutItems
- PutUsers

PutActionInteractions

Servicio: Amazon Personalize Events

Registra los datos de los eventos de interacción de acciones. Un evento de interacción de acciones es una interacción entre un usuario y una acción. Por ejemplo, un usuario que realiza una acción, como inscribirse en un programa para miembros o descargar su aplicación.

Para obtener información sobre cómo registrar interacciones de acción, consulte Registro de eventos de interacción de acciones. Para obtener más información sobre las acciones de un conjunto de datos de acciones, consulte Conjunto de datos de acciones.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

actionInteractions

Lista de los eventos de interacción de acciones de la sesión.

Tipo: Matriz de ActionInteraction objetos

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 10 elementos.

Obligatorio: sí

trackingId

ID de su rastreador de eventos de interacción de acciones. Cuando crea un conjunto de datos de interacciones de acción, Amazon Personalize crea un rastreador de eventos de interacciones de acción de forma automática. Para obtener más información, consulte ID del rastreador de eventos de interacciones de acción.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 409

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

PutActions

Servicio: Amazon Personalize Events

Agrega una o más acciones a un conjunto de datos de acciones. Para obtener más información, consulte Importación de acciones de forma individual.

Sintaxis de la solicitud

```
POST /actions HTTP/1.1
Content-type: application/json

{
   "actions": [
      {
         "actionId": "string",
         "properties": "string"
      }
   ],
   "datasetArn": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

actions

Lista de datos de acciones.

Tipo: Matriz de Action objetos

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 10 elementos.

Obligatorio: sí

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos de acciones al que está agregando las acciones.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 409

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2

- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

PutEvents

Servicio: Amazon Personalize Events

Registra los datos de los eventos de interacción de elementos. Para obtener más información, consulte Registro de eventos de interacción de elementos.

> **Note**
>
> Si usas una AWS Lambda función para llamar a la PutEvents operación, el rol de ejecución de la función debe tener permiso para realizar la `personalize:PutEvents` acción con el * comodín del `Resource` elemento.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

eventList

Una lista de datos de eventos de la sesión.

Tipo: Matriz de Event objetos

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 10 elementos.

Obligatorio: sí

sessionId

ID de sesión asociado a la visita del usuario. La aplicación genera el sessionId cuando un usuario visita por primera vez su sitio web o utiliza su aplicación. Amazon Personalize usa el sessionId para asociar eventos con el usuario antes de que inicie sesión. Para obtener más información, consulte Registro de eventos de interacción de elementos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

trackingId

ID de seguimiento del evento. El ID se genera mediante una llamada a la CreateEventTracker API.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

userId

El usuario asociado al evento.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

PutItems

Servicio: Amazon Personalize Events

Agrega uno o más elementos a un conjunto de datos de elementos. Para obtener más información, consulte Importación de elementos de forma individual.

Sintaxis de la solicitud

```
POST /items HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "items": [
      {
         "itemId": "string",
         "properties": "string"
      }
   ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos de elementos donde se va a añadir el elemento o elementos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

items

Una lista de datos de los elementos.

Tipo: Matriz de Item objetos

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 10 elementos.

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 409

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

PutUsers

Servicio: Amazon Personalize Events

Añade uno o más usuarios a un conjunto de datos de usuarios. Para obtener más información, consulte Importación de usuarios de forma individual.

Sintaxis de la solicitud

```
POST /users HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "users": [
      {
         "properties": "string",
         "userId": "string"
      }
   ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos de usuarios donde se va a añadir el usuario o usuarios.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

users

Una lista de datos de los usuarios.

Tipo: Matriz de User objetos

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 10 elementos.

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceInUseException

El recurso especificado está en uso.

Código de estado HTTP: 409

ResourceNotFoundException

No se ha encontrado el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

Versión ejecutable de Amazon Personalize

Versión ejecutable de Amazon Personalize admite las siguientes acciones:

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

GetActionRecommendations

Servicio: Amazon Personalize Runtime

Devuelve una lista de acciones recomendadas que se ordenan en orden descendente por puntuación de predicción. Use la API `GetActionRecommendations` si tiene una campaña personalizada que implementa una versión de la solución entrenada con una receta PERSONALIZED_ACTIONS.

Para obtener más información sobre las recetas PERSONALIZED_ACTIONS, consulte Recetas PERSONALIZED_ACTIONS. Para obtener más información sobre cómo obtener recomendaciones de acciones, consulte Obtención de recomendaciones de acciones.

Sintaxis de la solicitud

```
POST /action-recommendations HTTP/1.1
Content-type: application/json

{
   "campaignArn": "string",
   "filterArn": "string",
   "filterValues": {
      "string" : "string"
   },
   "numResults": number,
   "userId": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña que se va a utilizar para obtener recomendaciones de acciones. Esta campaña debe desplegar una versión de la solución entrenada con una receta PERSONALIZED_ACTIONS.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterArn

ARN del filtro que se aplicará a las recomendaciones devueltas. Para obtener más información, consulte Filtrado de recomendaciones.

Cuando utilice este parámetro, asegúrese de que el recurso de filtro sea `ACTIVE`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterValues

Los valores que se van a utilizar al filtrar las recomendaciones. Para cada parámetro de marcador de posición de la expresión de filtro, proporcione el nombre del parámetro (respetando mayúsculas y minúsculas) como clave y los valores del filtro como valor correspondiente. Separe los valores múltiples de un parámetro con una coma.

En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir acciones, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un elemento `EXCLUDE` para excluir acciones, puede omitir `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

Para obtener más información, consulte Filtrado de recomendaciones y segmentos de usuario.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 25 elementos.

Limitaciones de longitud de clave: longitud máxima de 50 caracteres.

Patrón de clave: `[A-Za-z0-9_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

numResults

El número de resultados que se devolverá. El valor predeterminado es 5. El máximo es 100.

Tipo: entero

Rango válido: valor mínimo de 0.

Obligatorio: no

userId

ID del usuario para el que se van a proporcionar recomendaciones de acciones.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-type: application/json

{
   "actionList": [
      {
         "actionId": "string",
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

actionList

Lista de recomendaciones de acciones en orden descendente por puntuación de predicción. Puede haber un máximo de 100 acciones en la lista. Para obtener información sobre las puntuaciones de las acciones, consulte Cómo funciona la puntuación de recomendaciones de acciones.

Tipo: matriz de objetos PredictedAction

recommendationId

El identificador de la recomendación.

Tipo: cadena

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

El recurso especificado no existe.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

GetPersonalizedRanking

Servicio: Amazon Personalize Runtime

Re-ranks una lista de artículos recomendados para un usuario determinado. Se considera que el primer elemento de la lista es el que puede ser el más interesante para el usuario.

> **Note**
>
> La solución relacionada con la campaña se debe haber creado utilizando una receta del tipo PERSONALIZED_RANKING.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña que se va a utilizar para generar la clasificación personalizada.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: sí

context

Los metadatos contextuales que se utilizarán para obtener recomendaciones. Los metadatos contextuales incluyen cualquier información de interacción que pueda ser relevante a la hora de obtener las recomendaciones de un usuario, como la ubicación actual del usuario o el tipo de dispositivo.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 150 elementos.

Limitaciones de longitud de clave: longitud máxima de 150 caracteres.

Patrón de clave: `[A-Za-z\d_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

filterArn

El nombre de recurso de Amazon (ARN) de un filtro que ha creado para incluir o excluir elementos de las recomendaciones para un usuario determinado. Para obtener más información, consulte Filtrado de recomendaciones.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterValues

Los valores que se van a utilizar al filtrar las recomendaciones. Para cada parámetro de marcador de posición de la expresión de filtro, proporcione el nombre del parámetro (respetando mayúsculas y minúsculas) como clave y los valores del filtro como valor correspondiente. Separe los valores múltiples de un parámetro con una coma.

En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir elementos, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un objeto `EXCLUDE` para excluir elementos, puede omitir la expresión `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

Para obtener más información, consulte Filtrado de recomendaciones.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 25 elementos.

Limitaciones de longitud de clave: longitud máxima de 50 caracteres.

Patrón de clave: `[A-Za-z0-9_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

inputList

Una lista de elementos (por `itemId`) para clasificar. Si un elemento no estaba incluido en el conjunto de datos de entrenamiento, se añade al final de la lista reclasificada. Si incluye metadatos en las recomendaciones, el máximo es 50. De lo contrario, el máximo es 500.

Tipo: matriz de cadenas

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

metadataColumns

Si ha habilitado los metadatos en las recomendaciones al crear o actualizar la campaña, especifique las columnas de metadatos del conjunto de datos de elementos que se van a incluir en la clasificación personalizada. La clave de asignación es `ITEMS` y el valor es una lista de

nombres de columnas de su conjunto de datos de elementos. El número máximo de columnas que puede proporcionar es 10.

Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Habilitación de los metadatos en las recomendaciones para una campaña.

Tipo: mapa de cadena a matriz de cadenas

Entradas de mapa: número máximo de 1 elemento.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Miembros de la matriz: número máximo de 99 elementos.

Limitaciones de longitud: longitud máxima de 150 caracteres.

Obligatorio: no

userId

El usuario para el que quiere que la campaña proporcione una clasificación personalizada.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-type: application/json

{
   "personalizedRanking": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
```

```
      }
   ],
   "recommendationId": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

personalizedRanking

Una lista de elementos por orden de interés más probable para el usuario. El máximo es 500.

Tipo: matriz de objetos PredictedItem

recommendationId

El identificador de la recomendación.

Tipo: cadena

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

El recurso especificado no existe.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2

- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

GetRecommendations

Servicio: Amazon Personalize Runtime

Devuelve una lista de elementos recomendados. Para las campañas, se requiere el nombre del recurso de Amazon (ARN) de la campaña y la entrada requerida de usuario y elemento depende del tipo de receta utilizado para crear la solución que respalda la campaña, de la siguiente manera:

- USER_PERSONALIZATION: `userId` obligatorio, `itemId` no utilizado
- RELATED_ITEMS: `itemId` obligatorio, `userId` no utilizado

> **Note**
>
> Las campañas respaldadas por una solución creada con una receta del tipo PERSONALIZED_RANKING utilizan la API GetPersonalizedRanking.

Para los recomendadores, se requiere el ARN del recomendador y la entrada requerida de usuario y elemento dependen del caso de uso (receta basada en el dominio) que respalda al recomendador. Para obtener información sobre los requisitos de los casos de uso, consulte Elección de los casos de uso de un recomendador.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña que se va a utilizar para obtener recomendaciones.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

context

Los metadatos contextuales que se utilizarán para obtener recomendaciones. Los metadatos contextuales incluyen cualquier información de interacción que pueda ser relevante a la hora de obtener las recomendaciones de un usuario, como la ubicación actual del usuario o el tipo de dispositivo.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 150 elementos.

Limitaciones de longitud de clave: longitud máxima de 150 caracteres.

Patrón de clave: `[A-Za-z\d_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

filterArn

ARN del filtro que se aplicará a las recomendaciones devueltas. Para obtener más información, consulte Filtrado de recomendaciones.

Cuando utilice este parámetro, asegúrese de que el recurso de filtro sea `ACTIVE`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterValues

Los valores que se van a utilizar al filtrar las recomendaciones. Para cada parámetro de marcador de posición de la expresión de filtro, proporcione el nombre del parámetro (respetando mayúsculas y minúsculas) como clave y los valores del filtro como valor correspondiente. Separe los valores múltiples de un parámetro con una coma.

En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir elementos, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un objeto `EXCLUDE` para excluir elementos, puede omitir la expresión `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

Para obtener más información, consulte Filtrado de recomendaciones y segmentos de usuario.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 25 elementos.

Limitaciones de longitud de clave: longitud máxima de 50 caracteres.

Patrón de clave: `[A-Za-z0-9_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

itemId

El ID del elemento para el que se van a ofrecer recomendaciones.

Necesario para el tipo de receta `RELATED_ITEMS`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

metadataColumns

Si ha habilitado los metadatos en las recomendaciones al crear o actualizar la campaña o el recomendador, especifique las columnas de metadatos del conjunto de datos de elementos que se van a incluir en las recomendaciones de elementos. La clave de asignación es `ITEMS` y el valor es una lista de nombres de columnas de su conjunto de datos de elementos. El número máximo de columnas que puede proporcionar es 10.

Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Habilitación de los metadatos en las recomendaciones para una campaña. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador.

Tipo: mapa de cadena a matriz de cadenas

Entradas de mapa: número máximo de 1 elemento.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Miembros de la matriz: número máximo de 99 elementos.

Limitaciones de longitud: longitud máxima de 150 caracteres.

Obligatorio: no

numResults

El número de resultados que se devolverá. El valor predeterminado es 25. Si incluye metadatos en las recomendaciones, el máximo es 50. De lo contrario, el máximo es 500.

Tipo: entero

Rango válido: valor mínimo de 0.

Obligatorio: no

promotions

Las promociones que se aplican a la solicitud de recomendación. Una promoción define reglas empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados.

Tipo: Matriz de objetos Promotion

Miembros de la matriz: número máximo de 1 elemento.

Obligatorio: no

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador que se va a utilizar para obtener recomendaciones. Proporcione el ARN de un recomendador si creó un grupo de conjuntos de datos de dominio con un recomendador para un caso de uso de dominio.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

userId

El ID del usuario para el que se van a proporcionar recomendaciones.

Necesario para el tipo de receta `USER_PERSONALIZATION`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-type: application/json

{
   "itemList": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

itemList

Lista de recomendaciones ordenadas en orden descendente por puntuación de predicción. Puede haber un máximo de 500 elementos en la lista.

Tipo: matriz de objetos PredictedItem

recommendationId

El identificador de la recomendación.

Tipo: cadena

Errores

InvalidInputException

Proporcione un valor válido para el campo o el parámetro.

Código de estado HTTP: 400

ResourceNotFoundException

El recurso especificado no existe.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los AWS SDK específicos del idioma, consulta lo siguiente:

- AWS Interfaz de línea de comandos V2
- AWS SDK para.NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para V3 JavaScript
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby V3

Data Types

Amazon Personalize admite los siguientes tipos de datos:

- 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

Eventos de Amazon Personalize admite los siguientes tipos de datos:

- Action
- ActionInteraction
- Event
- Item
- MetricAttribution
- User

Versión ejecutable de Amazon Personalize admite los siguientes tipos de datos:

- PredictedAction
- PredictedItem
- Promotion

Amazon Personalize

Amazon Personalize admite los siguientes tipos de datos:

- 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

Servicio: Amazon Personalize

Describe un algoritmo personalizado.

Contenido

algorithmArn

El nombre de recurso de Amazon (ARN) del algoritmo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

algorithmImage

El URI del contenedor de Docker para la imagen del algoritmo.

Tipo: objeto AlgorithmImage

Obligatorio: no

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó el algoritmo.

Tipo: marca temporal

Obligatorio: no

defaultHyperParameterRanges

Especifica los hiperparámetros predeterminados, sus rangos y si se pueden ajustar. Se puede determinar el valor de un hiperparámetro ajustable durante la optimización de hiperparámetros (HPO).

Tipo: objeto DefaultHyperParameterRanges

Obligatorio: no

defaultHyperParameters

Especifica los hiperparámetros predeterminados.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

defaultResourceConfig

Especifica el número máximo predeterminado de trabajos de entrenamiento y trabajos de entrenamiento en paralelo.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el algoritmo.

Tipo: marca temporal

Obligatorio: no

name

El nombre del algoritmo

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

roleArn

El nombre de recurso de Amazon (ARN) del rol.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

trainingInputMode

Modo de entrada de entrenamiento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

AlgorithmImage

Servicio: Amazon Personalize

Describe una imagen de algoritmo.

Contenido

dockerURI

El URI del contenedor de Docker para la imagen del algoritmo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

name

El nombre de la imagen del algoritmo.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

AutoMLConfig

Servicio: Amazon Personalize

Cuando la solución ejecuta AutoML (`performAutoML` es true en CreateSolution), Amazon Personalize determina qué receta, de la lista especificada, optimiza la métrica dada. Amazon Personalize luego utiliza esa receta para la solución.

Contenido

metricName

La métrica que se optimizará.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

recipeList

La lista de recetas candidatas.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 100 elementos.

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

AutoMLResult

Servicio: Amazon Personalize

Cuando la solución ejecuta AutoML (`performAutoML` se cumple en CreateSolution), especifica la receta que optimiza mejor la métrica especificada.

Contenido

bestRecipeArn

El nombre de recurso de Amazon (ARN) de la mejor receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

AutoTrainingConfig

Servicio: Amazon Personalize

La configuración de entrenamiento automático que se debe usar cuando el valor de `performAutoTraining` es true.

Contenido

schedulingExpression

Especifica la frecuencia con la que se entrenan automáticamente las nuevas versiones de solución. Especifique una expresión de frecuencia con el formato rate(valor unidad). En “valor”, especifique un número comprendido entre 1 y 30. En “unidad”, especifique `day` o `days`. Por ejemplo, para crear automáticamente una nueva versión de solución cada cinco días, especifique `rate(5 days)`. El valor predeterminado es cada siete días.

Para obtener más información sobre el entrenamiento automático, consulte Creación y configuración de una solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 16.

Patrón: `rate\(\d+ days?\)`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchInferenceJob

Servicio: Amazon Personalize

Contiene información sobre un trabajo de inferencia por lotes.

Contenido

batchInferenceJobArn

El nombre de recurso de Amazon (ARN) del trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

batchInferenceJobConfig

Un mapa de cadena a cadena de los detalles de configuración de un trabajo de inferencia por lotes.

Tipo: objeto BatchInferenceJobConfig

Obligatorio: no

batchInferenceJobMode

El modo del trabajo.

Tipo: cadena

Valores válidos: `BATCH_INFERENCE | THEME_GENERATION`

Obligatorio: no

creationDateTime

La hora en la que se creó el trabajo de inferencia por lotes.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produjo un error en el trabajo de inferencia por lotes, indica el motivo.

Tipo: cadena

Requerido: no

filterArn

El ARN del filtro usado en el trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

jobInput

La ruta de Amazon S3 que lleva a los datos de entrada utilizados para generar el trabajo de inferencia por lotes.

Tipo: objeto BatchInferenceJobInput

Obligatorio: no

jobName

El nombre del trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

jobOutput

El bucket de Amazon S3 que contiene los datos de salida generados por el trabajo de inferencia por lotes.

Tipo: objeto BatchInferenceJobOutput

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó por última vez el trabajo de inferencia por lotes.

Tipo: marca temporal

Obligatorio: no

numResults

El número de recomendaciones generadas por el trabajo de inferencia por lotes. Este número incluye los mensajes de error generados por los registros de entrada con error.

Tipo: entero

Obligatorio: no

roleArn

El ARN del rol de Amazon Identity and Access Management (IAM) que solicitó el trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

solutionVersionArn

El nombre del recurso de Amazon (ARN) de la versión de solución desde la que se creó el trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de inferencia por lotes. El estado es uno de los siguientes valores:

- PENDIENTE
- IN PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

themeGenerationConfig

La configuración de generación de temas del trabajo.

Tipo: objeto ThemeGenerationConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchInferenceJobConfig

Servicio: Amazon Personalize

Los detalles de configuración de un trabajo de inferencia por lotes.

Contenido

itemExplorationConfig

Un mapa de cadena a cadena que especifica los hiperparámetros de configuración de la exploración, incluidos `explorationWeight` y `explorationItemAgeCutOff`, que desea utilizar para configurar la cantidad de exploración de elementos que Amazon Personalize utiliza al recomendar artículos. Consulte User-Personalization.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

rankingInfluence

Un mapa de los valores de influencia de la clasificación para la POPULARIDAD y la FRESCURA. Para cada clave, especifique un valor numérico entre 0,0 y 1,0 que determine la influencia que tiene ese factor de clasificación en las recomendaciones finales. Un valor más cercano a 1,0 da más peso al factor, mientras que un valor más cercano a 0,0 reduce su influencia. Si no se especifica, ambos valores predeterminados son 0.0.

Tipo: cadena al mapa doble

Claves válidas: `POPULARITY | FRESHNESS`

Rango válido: valor mínimo de 0. El valor máximo es de 1.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulta lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

BatchInferenceJobInput

Servicio: Amazon Personalize

La configuración de entrada de un trabajo de inferencia por lotes.

Contenido

s3DataSource

El URI de la ubicación de Amazon S3 que contiene los datos de entrada. El bucket de Amazon S3 debe estar en la misma región que el punto de conexión de API al que está llamando.

Tipo: objeto S3DataConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchInferenceJobOutput

Servicio: Amazon Personalize

Los parámetros de configuración de salida de un trabajo de inferencia por lotes.

Contenido

s3DataDestination

Información sobre el bucket de Amazon S3 en el que se almacena el resultado del trabajo de inferencia por lotes.

Tipo: objeto S3DataConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchInferenceJobSummary

Servicio: Amazon Personalize

Una versión truncada de BatchInferenceJob. La operación ListBatchInferenceJobs devuelve una lista de resúmenes de trabajos de inferencia por lotes.

Contenido

batchInferenceJobArn

El nombre de recurso de Amazon (ARN) del trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

batchInferenceJobMode

El modo del trabajo.

Tipo: cadena

Valores válidos: `BATCH_INFERENCE | THEME_GENERATION`

Obligatorio: no

creationDateTime

La hora en la que se creó el trabajo de inferencia por lotes.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produjo un error en el trabajo de inferencia por lotes, indica el motivo.

Tipo: cadena

Requerido: no

jobName

El nombre del trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó por última vez el trabajo de inferencia por lotes.

Tipo: marca temporal

Obligatorio: no

solutionVersionArn

El ARN de la versión de solución usada por el trabajo de inferencia por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de inferencia por lotes. El estado es uno de los siguientes valores:

- PENDIENTE
- IN PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchSegmentJob

Servicio: Amazon Personalize

Contiene información sobre un trabajo de segmento por lotes.

Contenido

batchSegmentJobArn

El nombre de recurso de Amazon (ARN) del trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

creationDateTime

La hora en la que se creó el trabajo de segmento por lotes.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produjo un error en el trabajo de segmento por lotes, indica el motivo.

Tipo: cadena

Requerido: no

filterArn

El ARN del filtro usado en el trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

jobInput

La ruta de Amazon S3 que lleva a los datos de entrada utilizados para generar el trabajo de segmento por lotes.

Tipo: objeto BatchSegmentJobInput

Obligatorio: no

jobName

El nombre del trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

jobOutput

El bucket de Amazon S3 que contiene los datos de salida generados por el trabajo de segmento por lotes.

Tipo: objeto BatchSegmentJobOutput

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó por última vez el trabajo de segmento de lotes.

Tipo: marca temporal

Obligatorio: no

numResults

El número de usuarios previsto que generará el trabajo de segmentación por lotes para cada línea de datos de entrada. El número máximo de usuarios por segmento es de 5 millones.

Tipo: entero

Obligatorio: no

roleArn

El ARN del rol de Amazon Identity and Access Management (IAM) que solicitó el trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución que el trabajo de segmento por lotes utilizó para generar segmentos por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de segmento por lotes. El estado es uno de los siguientes valores:

- PENDIENTE
- IN PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchSegmentJobInput

Servicio: Amazon Personalize

La configuración de entrada de un trabajo de segmento por lotes.

Contenido

s3DataSource

Los detalles de configuración de un bucket de entrada o salida de Amazon S3.

Tipo: objeto S3DataConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchSegmentJobOutput

Servicio: Amazon Personalize

Los parámetros de configuración de salida de un trabajo de segmento por lotes.

Contenido

s3DataDestination

Los detalles de configuración de un bucket de entrada o salida de Amazon S3.

Tipo: objeto S3DataConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

BatchSegmentJobSummary

Servicio: Amazon Personalize

Una versión truncada del tipo de datos BatchSegmentJob. La operación ListBatchSegmentJobs devuelve una lista de resúmenes de trabajos de segmentos por lotes.

Contenido

batchSegmentJobArn

El nombre de recurso de Amazon (ARN) del trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

creationDateTime

La hora en la que se creó el trabajo de segmento por lotes.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produjo un error en el trabajo de segmento por lotes, indica el motivo.

Tipo: cadena

Requerido: no

jobName

El nombre del trabajo de segmento por lotes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó por última vez el trabajo de segmento por lotes.

Tipo: marca temporal

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución que el trabajo de segmento por lotes utilizó para generar segmentos por lotes.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de segmento por lotes. El estado es uno de los siguientes valores:

- PENDIENTE
- IN PROGRESS
- ACTIVE
- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

Campaign

Servicio: Amazon Personalize

Un objeto que describe la implementación de una versión de solución. Para obtener información acerca de las campañas, consulte CreateCampaign.

Contenido

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

campaignConfig

Los detalles de la configuración de una campaña.

Tipo: objeto CampaignConfig

Obligatorio: no

creationDateTime

La fecha y la hora (en formato Unix) en que se creó la campaña.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en la campaña, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en formato Unix) en que se actualizó por última vez la campaña.

Tipo: marca temporal

Obligatorio: no

latestCampaignUpdate

Proporciona un resumen de las propiedades de una actualización de campaña. Para obtener una lista completa, llame a la API DescribeCampaign.

> **Note**
>
> El campo `latestCampaignUpdate` solo se devuelve cuando la campaña ha tenido al menos una llamada de `UpdateCampaign`.

Tipo: objeto CampaignUpdateSummary

Obligatorio: no

minProvisionedTPS

Especifica el número mínimo de transacciones aprovisionadas (recomendaciones) por segundo solicitadas. Un valor elevado de `minProvisionedTPS` aumentará su factura. Le recomendamos empezar con 1 para `minProvisionedTPS` (el valor predeterminado). Realice un seguimiento de su uso mediante las métricas de Amazon CloudWatch y aumente el valor de `minProvisionedTPS` según sea necesario.

Tipo: número entero

Rango válido: valor mínimo de 1.

Obligatorio: no

name

Nombre de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

solutionVersionArn

Nombre de recurso de Amazon (ARN) de la versión de solución que usa la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la campaña.

Una campaña puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

CampaignConfig

Servicio: Amazon Personalize

Los detalles de la configuración de una campaña.

Contenido

enableMetadataWithRecommendations

Indica si los metadatos con recomendaciones están habilitados para la campaña. Si está habilitada, puede especificar las columnas del conjunto de datos de elementos en su solicitud de recomendaciones. Amazon Personalize devuelve estos datos para cada elemento en la respuesta a la recomendación. Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Habilitación de los metadatos en las recomendaciones para una campaña.

Si habilita los metadatos en las recomendaciones, incurrirá en costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Tipo: Booleano

Obligatorio: no

itemExplorationConfig

Especifica los hiperparámetros de configuración de la exploración, incluidos `explorationWeight` y `explorationItemAgeCutOff`, que desea utilizar para configurar la cantidad de exploración de elementos que Amazon Personalize utiliza al recomendar artículos. Proporcione datos `itemExplorationConfig` únicamente si la solución utiliza la receta User-Personalization.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

rankingInfluence

Un mapa de los valores de influencia de la clasificación para la POPULARIDAD y la FRESCURA. Para cada clave, especifique un valor numérico entre 0,0 y 1,0 que determine la influencia que tiene ese factor de clasificación en las recomendaciones finales. Un valor más cercano a 1,0 da más peso al factor, mientras que un valor más cercano a 0,0 reduce su influencia. Si no se especifica, ambos valores predeterminados son 0.0.

Tipo: cadena al mapa doble

Claves válidas: `POPULARITY | FRESHNESS`

Rango válido: valor mínimo de 0. El valor máximo es de 1.

Obligatorio: no

syncWithLatestSolutionVersion

Indica si la campaña se actualiza de forma automática para usar la versión de solución (modelo entrenado) más reciente de una solución. Si especifica `True`, debe especificar el ARN de la solución para el parámetro `SolutionVersionArn`. Debe tener el formato `SolutionArn/$LATEST`. El valor predeterminado es `False` y debe actualizar manualmente la campaña para implementar la última versión de solución.

Para obtener más información sobre las actualizaciones de campaña automáticas, consulte Habilitación de las actualizaciones de campaña automáticas.

Tipo: Booleano

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulta lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

CampaignSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una campaña. Para obtener una lista completa, llame a la API DescribeCampaign.

Contenido

campaignArn

El nombre de recurso de Amazon (ARN) de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó la campaña.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en la campaña, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la campaña.

Tipo: marca temporal

Obligatorio: no

name

Nombre de la campaña.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado de la campaña.

Una campaña puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

CampaignUpdateSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una actualización de campaña. Para obtener una lista completa, llame a la API DescribeCampaign.

Contenido

campaignConfig

Los detalles de la configuración de una campaña.

Tipo: objeto CampaignConfig

Obligatorio: no

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó la actualización de la campaña.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en la actualización de la campaña, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la actualización de la campaña.

Tipo: marca temporal

Obligatorio: no

minProvisionedTPS

Especifica el número mínimo de transacciones aprovisionadas (recomendaciones) por segundo solicitadas que Amazon Personalize va a admitir.

Tipo: número entero

Rango válido: valor mínimo de 1.

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución implementada.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la actualización de la campaña.

Una actualización de campaña puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

CategoricalHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango de un hiperparámetro categórico.

Contenido

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

values

Una lista de las categorías del hiperparámetro.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 100 elementos.

Limitaciones de longitud: longitud máxima de 1000 caracteres.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ContinuousHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango de un hiperparámetro continuo.

Contenido

maxValue

El valor máximo permitido para el hiperparámetro.

Tipo: doble

Rango válido: valor mínimo de -1000000.

Obligatorio: no

minValue

El valor mínimo permitido para el hiperparámetro.

Tipo: doble

Rango válido: valor mínimo de -1000000.

Obligatorio: no

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

DataDeletionJob

Servicio: Amazon Personalize

Describe un trabajo que elimina por lotes todas las referencias a usuarios específicos de un grupo de conjuntos de datos de Amazon Personalize. Para obtener información sobre cómo crear un trabajo de eliminación de datos, consulte Eliminación de usuarios.

Contenido

creationDateTime

Fecha y hora de creación (en tiempo Unix) del trabajo de eliminación de datos.

Tipo: marca temporal

Obligatorio: no

dataDeletionJobArn

Nombre de recurso de Amazon (ARN) del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetGroupArn

Nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos del que el trabajo elimina registros.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

dataSource

Describe el origen de datos que contiene los datos que se cargarán en un conjunto de datos o la lista de registros que se eliminarán de Amazon Personalize.

Tipo: objeto DataSource

Obligatorio: no

failureReason

Si se produce un error en un trabajo de eliminación de datos, se indica el motivo.

Tipo: cadena

Requerido: no

jobName

Nombre del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

Fecha y hora (en tiempo Unix) en que se actualizó por última vez el trabajo de eliminación de datos.

Tipo: marca temporal

Obligatorio: no

numDeleted

Número de registros eliminados por un trabajo COMPLETED.

Tipo: entero

Obligatorio: no

roleArn

Nombre de recurso de Amazon (ARN) del rol de IAM que tiene permisos para leer el origen de datos de Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

status

Estado del trabajo de eliminación de datos.

Un trabajo de eliminación de datos puede tener uno de los siguientes estados:

- PENDING > IN_PROGRESS > COMPLETED -o- FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DataDeletionJobSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un trabajo de eliminación de datos. Para obtener una lista completa, llame a la operación de la API DescribeDataDeletionJob.

Contenido

creationDateTime

Fecha y hora de creación (en tiempo Unix) del trabajo de eliminación de datos.

Tipo: marca temporal

Obligatorio: no

dataDeletionJobArn

Nombre de recurso de Amazon (ARN) del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetGroupArn

Nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos en el que el trabajo ha eliminado registros.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produce un error en un trabajo de eliminación de datos, se indica el motivo.

Tipo: cadena

Requerido: no

jobName

Nombre del trabajo de eliminación de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

Fecha y hora (en tiempo Unix) en que se actualizó por última vez el trabajo de eliminación de datos.

Tipo: marca temporal

Obligatorio: no

status

Estado del trabajo de eliminación de datos.

Un trabajo de eliminación de datos puede tener uno de los siguientes estados:

- PENDING > IN_PROGRESS > COMPLETED -o- FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

Dataset

Servicio: Amazon Personalize

Proporciona metadatos para un conjunto de datos.

Contenido

creationDateTime

La fecha y hora de creación (en tiempo Unix) del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos para el que desea metadatos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetType

Uno de los valores siguientes:

- Interacciones
- Items

- Users
- Acciones
- Action_Interactions

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `^[A-Za-z_]+$`

Obligatorio: no

lastUpdatedDateTime

Una marca de tiempo que muestra cuándo se actualizó el conjunto de datos.

Tipo: marca temporal

Obligatorio: no

latestDatasetUpdate

Describe la última actualización del conjunto de datos.

Tipo: objeto DatasetUpdateSummary

Obligatorio: no

name

Nombre del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

schemaArn

El ARN del esquema asociado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del conjunto de datos.

Un conjunto de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

trackingId

ID del rastreador de eventos de un conjunto de datos de interacciones de acción. Este ID se especifica en la operación de API `PutActionInteractions`. Amazon Personalize lo usa para dirigir los nuevos datos al conjunto de datos de interacciones de acción en su grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulta lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

DatasetExportJob

Servicio: Amazon Personalize

Describe un trabajo que exporta un conjunto de datos a un bucket de Amazon S3. Para obtener más información, consulte CreateDatasetExportJob.

Un trabajo de exportación de conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Contenido

creationDateTime

La fecha y hora de creación (en tiempo Unix) del trabajo de exportación de un conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetExportJobArn

El nombre de recurso de Amazon (ARN) del trabajo de exportación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produce un error en un trabajo de exportación de un conjunto de datos, indica el motivo.

Tipo: cadena

Requerido: no

ingestionMode

Los datos que se van a exportar, en función de cómo los haya importado. Puede optar por exportar los datos `BULK` que haya importado mediante un trabajo de importación de conjuntos de datos, los datos `PUT` que haya importado de forma incremental (mediante las operaciones de consola, PutEvents, PutUsers y PutItems) o `ALL` para ambos tipos. El valor predeterminado es `PUT`.

Tipo: cadena

Valores válidos: `BULK | PUT | ALL`

Obligatorio: no

jobName

Nombre del trabajo de exportación.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

jobOutput

La ruta al bucket de Amazon S3 donde se almacena el resultado del trabajo. Por ejemplo:

```
s3://bucket-name/folder-name/
```

Tipo: objeto DatasetExportJobOutput

Obligatorio: no

lastUpdatedDateTime

La fecha y la hora (en tiempo de Unix) en que se actualizó por última vez el estado del trabajo de exportación del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

roleArn

El nombre de recurso de Amazon (ARN) del rol de servicio de IAM que tiene permisos para añadir datos al bucket de Amazon S3 de salida.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de exportación del conjunto de datos.

Un trabajo de exportación de conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetExportJobOutput

Servicio: Amazon Personalize

Los parámetros de configuración de salida de un trabajo de exportación de conjuntos de datos.

Contenido

s3DataDestination

Los detalles de configuración de un bucket de entrada o salida de Amazon S3.

Tipo: objeto S3DataConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetExportJobSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un trabajo de exportación de conjuntos de datos. Para obtener una lista completa, llame a la API DescribeDatasetExportJob.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que se creó el trabajo de exportación del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetExportJobArn

El nombre de recurso de Amazon (ARN) del trabajo de exportación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produce un error en un trabajo de exportación de un conjunto de datos, indica el motivo.

Tipo: cadena

Requerido: no

jobName

Nombre para el trabajo de exportación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

La fecha y la hora (en tiempo de Unix) en que se actualizó por última vez el estado del trabajo de exportación del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

status

El estado del trabajo de exportación del conjunto de datos.

Un trabajo de exportación de conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetGroup

Servicio: Amazon Personalize

Un grupo de conjunto de datos es una colección de conjuntos de datos relacionados (interacciones de elementos, usuarios, elementos, acciones, interacciones de acción). Puede crear un grupo de conjuntos de datos mediante una llamada a CreateDatasetGroup. A continuación, cree un conjunto de datos y agréguelo a un grupo de conjuntos de datos mediante una llamada a CreateDataset. El grupo de conjuntos de datos se utiliza para crear y entrenar una solución mediante una llamada a CreateSolution. Un grupo de conjuntos de datos solo puede contener un conjunto de datos de cada tipo.

Puede especificar una clave AWS Key Management Service (KMS) para cifrar los conjuntos de datos del grupo.

Contenido

creationDateTime

La fecha y hora de creación (en tiempo Unix) del grupo de conjuntos de datos.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

domain

El dominio de un grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

failureReason

Si se produce un error al crear un grupo de conjuntos de datos, indica el motivo.

Tipo: cadena

Requerido: no

kmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave AWS Key Management Service (KMS) que se utiliza para cifrar los conjuntos datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 2048 caracteres.

Patrón: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora de la última actualización (en tiempo Unix) del grupo de conjuntos de datos.

Tipo: marca temporal

Obligatorio: no

name

El nombre del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

roleArn

El ARN del rol (de IAM) AWS Identity and Access Management que tiene permisos para acceder a la clave AWS Key Management Service (KMS). La especificación de un rol de IAM solo es válida cuando también se especifica una clave de KMS.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: no

status

El estado actual del grupo de conjuntos de datos.

Un grupo de conjuntos de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetGroupSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un grupo de conjuntos de datos. Para obtener una lista completa, llame a la API DescribeDatasetGroup.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que se creó el grupo de conjuntos de datos.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

domain

El dominio de un grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

failureReason

Si se produce un error en un grupo de conjuntos de datos, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el grupo de conjuntos de datos.

Tipo: marca temporal

Obligatorio: no

name

El nombre del grupo de conjuntos de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del grupo de conjuntos de datos.

Un grupo de conjuntos de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

DatasetImportJob

Servicio: Amazon Personalize

Describe un trabajo que importa datos de entrenamiento de un origen de datos (bucket de Amazon S3) a un conjunto de datos de Amazon Personalize. Para obtener más información, consulte CreateDatasetImportJob.

Un trabajo de importación de conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING (CREACIÓN PENDIENTE) > CREATE IN_PROGRESS (CREACIÓN EN CURSO) > ACTIVE (ACTIVO) o CREATE FAILED (ERROR AL CREAR)

Contenido

creationDateTime

La fecha y hora de creación (en tiempo Unix) del trabajo de importación de un conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos que recibe los datos importados.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetImportJobArn

El ARN del trabajo de importación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

dataSource

El bucket de Amazon S3 que contiene los datos de entrenamiento que se importarán.

Tipo: objeto DataSource

Obligatorio: no

failureReason

Si se produce un error en un trabajo de importación de un conjunto de datos, indica el motivo.

Tipo: cadena

Requerido: no

importMode

El modo de importación utilizado por el trabajo de importación del conjunto de datos para importar nuevos registros.

Tipo: cadena

Valores válidos: `FULL | INCREMENTAL`

Obligatorio: no

jobName

El nombre del trabajo de importación.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el conjunto de datos.

Tipo: marca temporal

Obligatorio: no

publishAttributionMetricsToS3

Si el trabajo publica métricas en Amazon S3 para una atribución de métricas.

Tipo: Booleano

Obligatorio: no

roleArn

El ARN del rol de IAM que tiene permisos de lectura del origen de datos de Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado del trabajo de importación del conjunto de datos.

Un trabajo de importación del conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING (CREACIÓN PENDIENTE) > CREATE IN_PROGRESS (CREACIÓN EN CURSO) > ACTIVE (ACTIVO) o CREATE FAILED (ERROR AL CREAR)

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

DatasetImportJobSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un trabajo de importación de conjuntos de datos. Para obtener una lista completa, llame a la API DescribeDatasetImportJob.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que se creó el trabajo de importación del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetImportJobArn

El nombre de recurso de Amazon (ARN) del trabajo de importación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produce un error en un trabajo de importación de un conjunto de datos, indica el motivo.

Tipo: cadena

Requerido: no

importMode

El modo de importación que el trabajo de importación del conjunto de datos utilizó para actualizar los datos del conjunto de datos. Para obtener más información, consulte Actualización de datos masivos existentes.

Tipo: cadena

Valores válidos: `FULL | INCREMENTAL`

Obligatorio: no

jobName

Nombre del trabajo de importación del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

lastUpdatedDateTime

La fecha y la hora (en tiempo de Unix) en que se actualizó por última vez el estado del trabajo de importación del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

status

El estado del trabajo de importación del conjunto de datos.

Un trabajo de importación del conjunto de datos puede aparecer en uno de los siguientes estados:

- CREATE PENDING (CREACIÓN PENDIENTE) > CREATE IN_PROGRESS (CREACIÓN EN CURSO) > ACTIVE (ACTIVO) o CREATE FAILED (ERROR AL CREAR)

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetSchema

Servicio: Amazon Personalize

Describe el esquema de un conjunto de datos. Para obtener más información sobre los esquemas, consulte CreateSchema.

Contenido

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó el esquema.

Tipo: marca temporal

Obligatorio: no

domain

El dominio de un esquema que creó para un conjunto de datos en un grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el esquema.

Tipo: marca temporal

Obligatorio: no

name

El nombre del esquema.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

schema

El esquema.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 20 000.

Obligatorio: no

schemaArn

Nombre de recurso de Amazon (ARN) del esquema.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetSchemaSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un esquema de conjuntos de datos. Para obtener una lista completa, llame a la API DescribeSchema.

Contenido

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó el esquema.

Tipo: marca temporal

Obligatorio: no

domain

El dominio de un esquema que creó para un conjunto de datos en un grupo de conjuntos de datos de dominio.

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el esquema.

Tipo: marca temporal

Obligatorio: no

name

El nombre del esquema.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

schemaArn

Nombre de recurso de Amazon (ARN) del esquema.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DatasetSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un conjunto de datos. Para obtener una lista completa, llama a la DescribeDatasetAPI.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que se creó el conjunto de datos.

Tipo: marca temporal

Obligatorio: no

datasetArn

El nombre de recurso de Amazon (ARN) del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

datasetType

El tipo del conjunto de datos. Uno de los valores siguientes:

- Interacciones
- Items
- Users
- Event-Interactions

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `^[A-Za-z_]+$`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el conjunto de datos.

Tipo: marca temporal

Obligatorio: no

name

Nombre del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del conjunto de datos.

Un conjunto de datos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulta lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

DatasetUpdateSummary

Servicio: Amazon Personalize

Describe una actualización de un conjunto de datos.

Contenido

creationDateTime

La fecha y hora de creación (en tiempo Unix) de la actualización del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en la actualización de un conjunto de datos, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora de la última actualización (en tiempo Unix) del conjunto de datos.

Tipo: marca temporal

Obligatorio: no

schemaArn

El nombre de recurso de Amazon (ARN) del esquema que sustituyó al esquema anterior del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la actualización del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DataSource

Servicio: Amazon Personalize

Describe el origen de datos que contiene los datos que se cargarán en un conjunto de datos o la lista de registros que se eliminarán de Amazon Personalize.

Contenido

dataLocation

En los trabajos de importación de conjuntos de datos, la ruta al bucket de Amazon S3 donde se almacenan los datos que desea cargar en el conjunto de datos. En los trabajos de eliminación de datos, la ruta al bucket de Amazon S3 que almacena la lista de registros que se eliminarán.

Por ejemplo:

`s3://bucket-name/folder-name/fileName.csv`

Si los archivos CSV se encuentran en un bucket de Amazon S3 y desea que en el trabajo de importación o de eliminación de datos se tengan en cuenta varios archivos, puede especificar la ruta a la carpeta. Con una tarea de eliminación de datos, Amazon Personalize utiliza todos los archivos de la carpeta y de cualquier subcarpeta. Use la siguiente sintaxis con una `/` después del nombre de la carpeta:

`s3://bucket-name/folder-name/`

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `(s3|http|https)://.+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

DefaultCategoricalHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango predeterminado de un hiperparámetro categórico y si el hiperparámetro se puede ajustar. Se puede determinar el valor de un hiperparámetro ajustable durante la optimización de hiperparámetros (HPO).

Contenido

isTunable

Si el hiperparámetro se puede ajustar.

Tipo: Booleano

Obligatorio: no

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

values

Una lista de las categorías del hiperparámetro.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 100 elementos.

Limitaciones de longitud: longitud máxima de 1000 caracteres.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DefaultContinuousHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango predeterminado de un hiperparámetro continuo y si el hiperparámetro se puede ajustar. Se puede determinar el valor de un hiperparámetro ajustable durante la optimización de hiperparámetros (HPO).

Contenido

isTunable

Si el hiperparámetro se puede ajustar.

Tipo: Booleano

Obligatorio: no

maxValue

El valor máximo permitido para el hiperparámetro.

Tipo: doble

Rango válido: valor mínimo de -1000000.

Obligatorio: no

minValue

El valor mínimo permitido para el hiperparámetro.

Tipo: doble

Rango válido: valor mínimo de -1000000.

Obligatorio: no

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DefaultHyperParameterRanges

Servicio: Amazon Personalize

Especifica los hiperparámetros y sus rangos predeterminados. Los hiperparámetros pueden ser categóricos o continuos o tener valores enteros.

Contenido

categoricalHyperParameterRanges

Los hiperparámetros categóricos y sus rangos predeterminados.

Tipo: matriz de objetos DefaultCategoricalHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

continuousHyperParameterRanges

Los hiperparámetros continuos y sus rangos predeterminados.

Tipo: matriz de objetos DefaultContinuousHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

integerHyperParameterRanges

Los hiperparámetros con valores enteros y sus rangos predeterminados.

Tipo: matriz de objetos DefaultIntegerHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

DefaultIntegerHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango predeterminado de un hiperparámetro con valores enteros y si el hiperparámetro se puede ajustar. Se puede determinar el valor de un hiperparámetro ajustable durante la optimización de hiperparámetros (HPO).

Contenido

isTunable

Indica si el hiperparámetro se puede ajustar.

Tipo: Booleano

Obligatorio: no

maxValue

El valor máximo permitido para el hiperparámetro.

Tipo: número entero

Rango válido: valor máximo de 1000000.

Obligatorio: no

minValue

El valor mínimo permitido para el hiperparámetro.

Tipo: número entero

Rango válido: valor mínimo de -1000000.

Obligatorio: no

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

EventParameters

Servicio: Amazon Personalize

Describe los parámetros de evento que se utilizan para en la creación de soluciones.

Contenido

eventType

El nombre del tipo de evento que se debe tener en cuenta para la creación de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

eventValueThreshold

El umbral del tipo de evento. Para crear la solución, solo se tendrán en cuenta los eventos con un valor superior o igual a este umbral.

Tipo: Doble

Obligatorio: no

weight

La ponderación del tipo de evento. Una ponderación más alta significa que el tipo de evento tiene mayor importancia para la solución creada.

Tipo: doble

Rango válido: valor mínimo de 0. El valor máximo es de 1.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

EventsConfig

Servicio: Amazon Personalize

Describe la configuración de los eventos, que se utilizan en la creación de soluciones.

Contenido

eventParametersList

Una lista de parámetros de eventos, que incluye los tipos de evento y los umbrales y ponderaciones de los valores de evento.

Tipo: matriz de objetos EventParameters

Miembros de la matriz: número máximo de 10 elementos.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

EventTracker

Servicio: Amazon Personalize

Proporciona información sobre un rastreador de eventos.

Contenido

accountId

La cuenta AWS propietaria del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

creationDateTime

La fecha y la hora (en formato Unix) en que se creó el rastreador de eventos.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que recibe los datos del evento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

eventTrackerArn

El ARN del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el rastreador de eventos.

Tipo: marca temporal

Obligatorio: no

name

El nombre del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del rastreador de eventos.

Un rastreador de eventos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

trackingId

ID del rastreador de eventos. Incluya este ID en las solicitudes a la API PutEvents.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

EventTrackerSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de un rastreador de eventos. Para obtener una lista completa, llame a la API DescribeEventTracker.

Contenido

creationDateTime

La fecha y la hora (en hora Unix) en que se creó el rastreador de eventos.

Tipo: marca temporal

Obligatorio: no

eventTrackerArn

El nombre de recurso de Amazon (ARN) del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez el rastreador de eventos.

Tipo: marca temporal

Obligatorio: no

name

El nombre del rastreador de eventos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del rastreador de eventos.

Un rastreador de eventos puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

FeatureTransformation

Servicio: Amazon Personalize

Proporciona información sobre la transformación de características. La transformación de características es el proceso de modificar los datos de entrada sin procesar para que tengan un formato más adecuado para el entrenamiento de modelos.

Contenido

creationDateTime

La fecha y la hora de creación (en tiempo Unix) de la transformación de la característica.

Tipo: marca temporal

Obligatorio: no

defaultParameters

Proporciona los parámetros predeterminados para la transformación de características.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

featureTransformationArn

El nombre de recurso de Amazon (ARN) del objeto FeatureTransformation.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La fecha y la hora de la última actualización (en tiempo Unix) de la transformación de características.

Tipo: marca temporal

Obligatorio: no

name

El nombre de la transformación de características.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado de la transformación de características.

Un transformación de características puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

FieldsForThemeGeneration

Servicio: Amazon Personalize

Asignación de cadena a cadena de los detalles de configuración para la generación de temas.

Contenido

itemName

El nombre de la columna del conjunto de datos de elementos que almacena el nombre de cada elemento del conjunto de datos.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 150 caracteres.

Patrón: `[A-Za-z_][A-Za-z\d_]*`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Filter

Servicio: Amazon Personalize

Contiene información sobre un filtro de recomendaciones, incluidos su ARN, estado y expresión de filtro.

Contenido

creationDateTime

La hora a la que se creó el filtro.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El ARN del grupo de conjuntos de datos al que pertenece el filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produjo un error en el filtro, indica el motivo.

Tipo: cadena

Requerido: no

filterArn

El ARN del filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterExpression

Especifica el tipo de interacciones entre los elementos que se van a filtrar de los resultados de las recomendaciones. La expresión de filtro debe seguir reglas de formato específicas. Para obtener información sobre la estructura y la sintaxis de la expresión de filtro, consulte Expresiones de filtro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 2500 caracteres.

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó el filtro por última vez.

Tipo: marca temporal

Obligatorio: no

name

El nombre del filtro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

FilterSummary

Servicio: Amazon Personalize

Un breve resumen de los atributos de un filtro.

Contenido

creationDateTime

La hora a la que se creó el filtro.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El ARN del grupo de conjuntos de datos al que pertenece el filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produjo un error en el filtro, indica el motivo.

Tipo: cadena

Requerido: no

filterArn

El ARN del filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La hora en la que se actualizó el filtro por última vez.

Tipo: marca temporal

Obligatorio: no

name

El nombre del filtro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado del filtro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

HPOConfig

Servicio: Amazon Personalize

Describe las propiedades de la optimización de hiperparámetros (HPO).

Contenido

algorithmHyperParameterRanges

Los hiperparámetros y los rangos permitidos.

Tipo: objeto HyperParameterRanges

Obligatorio: no

hpoObjective

La métrica que optimizará durante la HPO.

> **Note**
>
> Amazon Personalize no admite la configuración de `hpoObjective` en este momento.

Tipo: objeto HPOObjective

Obligatorio: no

hpoResourceConfig

Describe la configuración de recursos para HPO.

Tipo: objeto HPOResourceConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

HPOObjective

Servicio: Amazon Personalize

La métrica que se optimizará durante la optimización de hiperparámetros (HPO).

> **Note**
>
> Amazon Personalize no admite la configuración de `hpoObjective` en este momento.

Contenido

metricName

El nombre de la métrica.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

metricRegex

Una expresión regular para buscar la métrica en los registros de trabajos de formación.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

type

El tipo de la métrica. Los valores válidos son `Maximize` y `Minimize`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

HPOResourceConfig

Servicio: Amazon Personalize

Describe la configuración de recursos para la optimización de hiperparámetros (HPO).

Contenido

maxNumberOfTrainingJobs

La cantidad máxima de trabajos de entrenamiento cuando se crea una versión de la solución. El valor máximo para `maxNumberOfTrainingJobs` es `40`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

maxParallelTrainingJobs

La cantidad máxima de trabajos de entrenamiento paralelos cuando se crea una versión de la solución. El valor máximo para `maxParallelTrainingJobs` es `10`.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

HyperParameterRanges

Servicio: Amazon Personalize

Especifica los hiperparámetros y sus rangos. Los hiperparámetros pueden ser categóricos o continuos o tener valores enteros.

Contenido

categoricalHyperParameterRanges

Los hiperparámetros categóricos y sus rangos.

Tipo: matriz de objetos CategoricalHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

continuousHyperParameterRanges

Los hiperparámetros continuos y sus rangos.

Tipo: matriz de objetos ContinuousHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

integerHyperParameterRanges

Los hiperparámetros con valores enteros y sus rangos.

Tipo: matriz de objetos IntegerHyperParameterRange

Miembros de la matriz: número máximo de 100 elementos.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

IntegerHyperParameterRange

Servicio: Amazon Personalize

Proporciona el nombre y el rango de un hiperparámetro de valores enteros.

Contenido

maxValue

El valor máximo permitido para el hiperparámetro.

Tipo: número entero

Rango válido: valor máximo de 1000000.

Obligatorio: no

minValue

El valor mínimo permitido para el hiperparámetro.

Tipo: número entero

Rango válido: valor mínimo de -1000000.

Obligatorio: no

name

El nombre del hiperparámetro.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

MetricAttribute

Servicio: Amazon Personalize

Contiene información sobre una métrica de la que informa una atribución de métricas. Para obtener más información, consulte Medición del impacto de las recomendaciones.

Contenido

eventType

El tipo de evento de la métrica.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

expression

La expresión del atributo. Las funciones disponibles son `SUM()` o `SAMPLECOUNT()`. Para las funciones SUM(), proporcione el tipo de conjunto de datos (interacciones o elementos) y la columna que se va a sumar como parámetro. Por ejemplo, SUM(Items.PRICE).

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

metricName

El nombre de la métrica. El nombre le ayuda a identificar la métrica en Amazon CloudWatch o Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MetricAttribution

Servicio: Amazon Personalize

Contiene información sobre una atribución de métricas. La atribución de métricas crea informes sobre los datos que importa en Amazon Personalize. En función de cómo haya importado los datos, los informes se pueden ver en Amazon CloudWatch o Amazon S3. Para obtener más información, consulte Medición del impacto de las recomendaciones.

Contenido

creationDateTime

Fecha y hora de creación de la atribución de métricas.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de las atribuciones de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

El motivo del error de la atribución de métricas.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

Fecha y hora de la última actualización de la atribución de métricas.

Tipo: marca temporal

Obligatorio: no

metricAttributionArn

El nombre de recurso de Amazon (ARN) de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

metricsOutputConfig

La configuración de salida de la atribución de métricas.

Tipo: objeto MetricAttributionOutput

Obligatorio: no

name

El nombre de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MetricAttributionOutput

Servicio: Amazon Personalize

Los detalles de configuración de salida para la atribución de métricas.

Contenido

roleArn

El nombre de recurso de Amazon (ARN) del rol de servicio de IAM que tiene permisos para añadir datos al bucket de Amazon S3 de salida y añadir métricas a Amazon CloudWatch. Para obtener más información, consulte Medición del impacto de las recomendaciones.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Obligatorio: sí

s3DataDestination

Los detalles de configuración de un bucket de entrada o salida de Amazon S3.

Tipo: objeto S3DataConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MetricAttributionSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una atribución de métricas. Para obtener una lista completa, llame a DescribeMetricAttribution.

Contenido

creationDateTime

Fecha y hora de creación de la atribución de métricas.

Tipo: marca temporal

Obligatorio: no

failureReason

El motivo del error de la atribución de métricas.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

Fecha y hora de la última actualización de la atribución de métricas.

Tipo: marca temporal

Obligatorio: no

metricAttributionArn

El nombre de recurso de Amazon (ARN) de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

name

El nombre de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

status

El estado de la atribución de métricas.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

OptimizationObjective

Servicio: Amazon Personalize

Describe el objetivo adicional de la solución, como maximizar los minutos de streaming o aumentar los ingresos. Para obtener más información, consulte Optimización de una solución.

Contenido

itemAttribute

La columna de metadatos numéricos de un conjunto de datos de elementos relacionada con el objetivo de optimización. Por ejemplo, VIDEO_LENGTH (para maximizar los minutos de streaming) o PRICE (para maximizar los ingresos).

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 150 caracteres.

Obligatorio: no

objectiveSensitivity

Especifica cómo Amazon Personalize equilibra la importancia de su objetivo de optimización con la relevancia.

Tipo: cadena

Valores válidos: `LOW | MEDIUM | HIGH | OFF`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Recipe

Servicio: Amazon Personalize

Proporciona información sobre una receta. Cada receta proporciona un algoritmo que Amazon Personalize utiliza en el entrenamiento de modelos cuando se utiliza la operación CreateSolution.

Contenido

algorithmArn

El nombre de recurso de Amazon (ARN) del algoritmo que Amazon Personalize utiliza para entrenar el modelo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

creationDateTime

La fecha y la hora (en formato Unix) en que se creó la receta.

Tipo: marca temporal

Obligatorio: no

description

La descripción de la receta.

Tipo: cadena

Requerido: no

featureTransformationArn

El ARN del objeto FeatureTransformation.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en formato Unix) en que se actualizó por última vez la receta.

Tipo: marca temporal

Obligatorio: no

name

El nombre de la receta.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

recipeType

Uno de los valores siguientes:

- PERSONALIZED_RANKING
- RELATED_ITEMS
- USER_PERSONALIZATION

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

status

El estado de la receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

RecipeSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una receta. Para obtener una lista completa, llame a la API DescribeRecipe.

Contenido

creationDateTime

La fecha y la hora (en hora Unix) en que se creó la receta.

Tipo: marca temporal

Obligatorio: no

domain

El dominio de la receta (si la receta es un caso de uso de un grupo de conjuntos de datos de dominio).

Tipo: cadena

Valores válidos: `ECOMMERCE | VIDEO_ON_DEMAND`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la receta.

Tipo: marca temporal

Obligatorio: no

name

El nombre de la receta.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la receta.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Recommender

Servicio: Amazon Personalize

Describe un generador de recomendaciones para un grupo de conjuntos de datos de dominio. Crea un recomendador en un grupo de conjuntos de datos de dominio para un caso de uso de dominio específico (receta de dominio) y especifica el recomendador en una solicitud de GetRecommendations.

Contenido

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó el recomendador.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de dominio que contiene el recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

failureReason

Si se produce un error en el recomendador, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en formato Unix) en que se actualizó por última vez el recomendador.

Tipo: marca temporal

Obligatorio: no

latestRecommenderUpdate

Proporciona un resumen de las últimas actualizaciones del recomendador.

Tipo: objeto RecommenderUpdateSummary

Obligatorio: no

modelMetrics

Proporciona métricas de evaluación que ayudan a determinar el rendimiento de un recomendador. Para obtener más información, consulte Evaluación de un recomendador.

Tipo: cadena al mapa doble

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Obligatorio: no

name

El nombre del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta (caso de uso del grupo de conjuntos de datos de dominio) para el que se creó el recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

recommenderConfig

Los detalles de configuración del recomendador.

Tipo: objeto RecommenderConfig

Obligatorio: no

status

El estado del recomendador.

Un recomendador puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

RecommenderConfig

Servicio: Amazon Personalize

Los detalles de configuración del recomendador.

Contenido

enableMetadataWithRecommendations

Indica si los metadatos con recomendaciones están habilitados para el recomendador. Si está habilitada, puede especificar las columnas del conjunto de datos de elementos en su solicitud de recomendaciones. Amazon Personalize devuelve estos datos para cada elemento en la respuesta a la recomendación. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador.

Si habilita los metadatos en las recomendaciones, incurrirá en costos adicionales. Para obtener más información, consulte Precios de Amazon Personalize.

Tipo: Booleano

Obligatorio: no

itemExplorationConfig

Especifica los hiperparámetros de configuración de la exploración, incluidos `explorationWeight` y `explorationItemAgeCutOff`, que desea utilizar para configurar la cantidad de exploración de elementos que Amazon Personalize utiliza al recomendar artículos. Proporciona datos `itemExplorationConfig` solo si sus recomendadores generan recomendaciones personalizadas para un usuario (no elementos populares ni elementos similares).

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

minRecommendationRequestsPerSecond

Especifica el número mínimo de solicitudes de recomendación aprovisionadas solicitadas por segundo que Amazon Personalize admitirá. Un valor `minRecommendationRequestsPerSecond` elevado incrementará su factura. Le recomendamos empezar con 1 para `minRecommendationRequestsPerSecond` (el valor predeterminado). Realice un seguimiento de su uso mediante las métricas de Amazon CloudWatch y aumente `minRecommendationRequestsPerSecond` según sea necesario.

Tipo: número entero

Rango válido: valor mínimo de 1.

Obligatorio: no

trainingDataConfig

Especifica la configuración de los datos de entrenamiento que se utilizará al crear un recomendador de dominios.

Tipo: objeto TrainingDataConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

RecommenderSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades del recomendador.

Contenido

creationDateTime

La fecha y la hora (en tiempo Unix) en que se creó el recomendador.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos de dominio que contiene el recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en formato Unix) en que se actualizó por última vez el recomendador.

Tipo: marca temporal

Obligatorio: no

name

El nombre del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta (caso de uso del grupo de conjuntos de datos de dominio) para el que se creó el recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

recommenderArn

El nombre de recurso de Amazon (ARN) del recomendador.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

recommenderConfig

Los detalles de configuración del recomendador.

Tipo: objeto RecommenderConfig

Obligatorio: no

status

El estado del recomendador. Un recomendador puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

RecommenderUpdateSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una actualización de recomendador. Para obtener una lista completa, llame a la API DescribeRecommender.

Contenido

creationDateTime

La fecha y la hora (en formato Unix) en que se creó la actualización del recomendador.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en la actualización del recomendador, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la actualización del recomendador.

Tipo: marca temporal

Obligatorio: no

recommenderConfig

Los detalles de configuración de la actualización del recomendador.

Tipo: objeto RecommenderConfig

Obligatorio: no

status

El estado de la actualización del recomendador. Una actualización de recomendador puede tener uno de los siguientes estados:

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

S3DataConfig

Servicio: Amazon Personalize

Los detalles de configuración de un bucket de entrada o salida de Amazon S3.

Contenido

path

La ruta de archivo del bucket de Amazon S3.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `(s3|http|https)://.+`

Obligatorio: sí

kmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave AWS Key Management Service (KMS) que Amazon Personalize utiliza para cifrar o descifrar los archivos de entrada y salida.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 2048 caracteres.

Patrón: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Solution

Servicio: Amazon Personalize

> **Important**
>
> De forma predeterminada, todas las soluciones nuevas utilizan el entrenamiento automático. Con el entrenamiento automático, incurrirá en costos de entrenamiento mientras la solución esté activa. Para evitar costos innecesarios, cuando haya terminado, puede actualizar la solución para desactivar el entrenamiento automático. Para obtener información sobre los costos de entrenamiento, consulte Precios de Amazon Personalize.

Un objeto que proporciona información sobre una solución. Una solución incluye la receta personalizada, los parámetros personalizados y los modelos entrenados (versiones de solución) que Amazon Personalize utiliza para generar recomendaciones.

Después de crear una solución, no puede cambiar la configuración. Si necesita realizar cambios, puede clonar la solución con la consola de Amazon Personalize o crear una nueva.

Contenido

autoMLResult

Cuando `performAutoML` es verdadero, especifica la mejor receta encontrada.

Tipo: objeto AutoMLResult

Obligatorio: no

creationDateTime

La fecha y hora de creación (en tiempo Unix) de la solución.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que proporciona los datos de entrenamiento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

eventType

El tipo de evento (por ejemplo, “clic” o “me gusta”) que se utiliza para entrenar el modelo. Si no se proporciona `eventType`, Amazon Personalize utiliza todas las interacciones para el entrenamiento con la misma ponderación, independientemente del tipo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la solución.

Tipo: marca temporal

Obligatorio: no

latestSolutionUpdate

Proporciona un resumen de las últimas actualizaciones de la solución.

Tipo: objeto SolutionUpdateSummary

Obligatorio: no

latestSolutionVersion

Describe la versión más reciente de la solución, incluidos el estado y el ARN.

Tipo: objeto SolutionVersionSummary

Obligatorio: no

name

El nombre de la solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

performAutoML

> **Important**
>
> No le recomendamos habilitar el machine learning automático. En lugar de eso, puede emparejar su caso de uso con las recetas de Amazon Personalize disponibles. Para obtener más información, consulte Determining your use case. (Determinación de su caso de uso).

Si se cumple, Amazon Personalize realizará una búsqueda de la mejor receta USER_PERSONALIZATION de la lista especificada en la configuración de la solución (`recipeArn` no debe especificarse). Si no se cumple, Amazon Personalize utilizará `recipeArn` para el entrenamiento.

Tipo: Booleano

Obligatorio: no

performAutoTraining

Especifica si la solución crea versiones de solución de forma automática. El valor predeterminado es `True` y la solución crea automáticamente nuevas versiones de solución cada siete días.

Para obtener más información sobre el entrenamiento automático, consulte Creación y configuración de una solución.

Tipo: Booleano

Obligatorio: no

performHPO

Si se debe realizar la optimización de hiperparámetros (HPO) en la receta que se ha elegido. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

performIncrementalUpdate

Un valor booleano que indica si se realizan actualizaciones de entrenamiento incrementales en el modelo. Cuando está activado, permite que el modelo aprenda de los nuevos datos con mayor frecuencia sin necesidad de volver a capacitarse por completo, lo que permite una personalización prácticamente en tiempo real. Este parámetro solo es compatible con las soluciones que utilizan la receta de similitud semántica

Tipo: Booleano

Obligatorio: no

recipeArn

El ARN de la receta que se ha utilizado para crear la solución. Esto es obligatorio cuando `performAutoML` es falso.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionArn

El ARN de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionConfig

Describe las propiedades de configuración de la solución.

Tipo: objeto SolutionConfig

Obligatorio: no

status

El estado de la acción.

Una solución puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulte lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

SolutionConfig

Servicio: Amazon Personalize

Describe las propiedades de configuración de la solución.

Contenido

algorithmHyperParameters

Muestra los hiperparámetros del algoritmo y sus valores.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

autoMLConfig

El objeto AutoMLConfig que contiene una lista de recetas para buscar cuando se ejecuta AutoML.

Tipo: objeto AutoMLConfig

Obligatorio: no

autoTrainingConfig

Especifica la configuración de entrenamiento automático que se utilizará.

Tipo: objeto AutoTrainingConfig

Obligatorio: no

eventsConfig

Describe la configuración de un evento, que incluye una lista de parámetros de evento. Puede especificar hasta 10 parámetros de evento. Los eventos se utilizan en la creación de soluciones.

Tipo: objeto EventsConfig

Obligatorio: no

eventValueThreshold

Para entrenar un modelo, solo se utilizan eventos cuyo valor sea mayor o igual que el umbral.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

featureTransformationParameters

Muestra los parámetros de transformación de características.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

hpoConfig

Describe las propiedades de la optimización de hiperparámetros (HPO).

Tipo: objeto HPOConfig

Obligatorio: no

optimizationObjective

Describe el objetivo adicional de la solución, como maximizar los minutos de streaming o aumentar los ingresos. Para obtener más información, consulte Optimización de una solución.

Tipo: objeto OptimizationObjective

Obligatorio: no

trainingDataConfig

La configuración de los datos de entrenamiento que se utilizará al crear una versión de solución personalizada (modelo entrenado).

Tipo: objeto TrainingDataConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SolutionSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una solución. Para obtener una lista completa, llame a la API DescribeSolution.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que creó la solución.

Tipo: marca temporal

Obligatorio: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la solución.

Tipo: marca temporal

Obligatorio: no

name

El nombre de la solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

recipeArn

El nombre de recurso de Amazon (ARN) de la receta que utiliza la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionArn

El nombre de recurso de Amazon (ARN) de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la acción.

Una solución puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SolutionUpdateConfig

Servicio: Amazon Personalize

Los detalles de configuración de la actualización de la solución.

Contenido

autoTrainingConfig

La configuración de entrenamiento automático que se debe usar cuando el valor de `performAutoTraining` es true.

Tipo: objeto AutoTrainingConfig

Obligatorio: no

eventsConfig

Describe la configuración de un evento, que incluye una lista de parámetros de evento. Puede especificar hasta 10 parámetros de evento. Los eventos se utilizan en la creación de soluciones.

Tipo: objeto EventsConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SolutionUpdateSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una actualización de solución. Para obtener una lista completa, llama a la DescribeSolutionAPI.

Contenido

creationDateTime

La fecha y la hora (en formato Unix) en que se creó la actualización de solución.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en una actualización de solución, se indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

Fecha y hora (en tiempo Unix) en que se actualizó por última vez la actualización de solución.

Tipo: marca temporal

Obligatorio: no

performAutoTraining

Especifica si la solución crea automáticamente versiones de solución.

Tipo: Booleano

Obligatorio: no

performIncrementalUpdate

Un valor booleano que indica si se realizan actualizaciones de entrenamiento incrementales en el modelo. Cuando está activado, el entrenamiento incremental se realiza para las versiones de la solución con campañas activas y permite que el modelo aprenda de los nuevos datos

con mayor frecuencia sin necesidad de volver a capacitarse por completo, lo que permite una personalización casi en tiempo real. Este parámetro solo es compatible con las soluciones que utilizan la receta de similitud semántica. El valor predeterminado es `true`.

Ten en cuenta que es posible que determinadas puntuaciones y atributos (como las actualizaciones de popularidad y frescura de los artículos utilizadas para influir en la clasificación en la aws-semantic-similarity receta) no se actualicen hasta que se realice el siguiente entrenamiento completo.

Tipo: Booleano

Obligatorio: no

solutionUpdateConfig

Detalles de configuración de la solución.

Tipo: objeto SolutionUpdateConfig

Obligatorio: no

status

Estado de la actualización de solución. Una actualización de solución puede tener uno de los siguientes estados:

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulta lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

SolutionVersion

Servicio: Amazon Personalize

Objeto que proporciona información sobre una versión específica de una solución en un grupo de conjuntos de datos personalizado.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que creó la versión de la solución.

Tipo: marca temporal

Obligatorio: no

datasetGroupArn

El nombre de recurso de Amazon (ARN) del grupo de conjuntos de datos que proporciona los datos de entrenamiento.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

eventType

El tipo de evento (por ejemplo, “clic” o “me gusta”) que se utiliza para entrenar el modelo.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

failureReason

Si se produce un error en el entrenamiento de una versión de solución, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la solución.

Tipo: marca temporal

Obligatorio: no

name

El nombre de la versión de la solución.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

performAutoML

Si es cierto, Amazon Personalize busca la receta más óptima según la configuración de la solución. Cuando es false (valor predeterminado), Amazon Personalize utiliza `recipeArn`.

Tipo: Booleano

Obligatorio: no

performHPO

Si se debe realizar la optimización de hiperparámetros (HPO) en la receta que se ha elegido. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

performIncrementalUpdate

Si la versión de la solución debe realizar una actualización incremental. Si se establece en true, el entrenamiento procesará solo los datos que hayan cambiado desde el último entrenamiento, de forma similar a cuando TrainingMode está configurado en UPDATE. Esto solo se puede usar con las versiones de la solución que utilizan la receta de personalización del usuario.

Tipo: Booleano

Obligatorio: no

recipeArn

El ARN de la receta que se ha utilizado en la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionArn

El ARN de la solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

solutionConfig

Describe las propiedades de configuración de la solución.

Tipo: objeto SolutionConfig

Obligatorio: no

solutionVersionArn

El ARN de la versión de solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la versión de solución.

Una versión de la solución puede tener uno de los siguientes estados:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING
- CREATE STOPPED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

trainingHours

El tiempo utilizado para entrenar el modelo. Se le facturará por el tiempo que se tarda en entrenar un modelo. Este campo solo está visible después de que Amazon Personalize entrene correctamente un modelo.

Tipo: Doble

Rango válido: valor mínimo de 0.

Obligatorio: no

trainingMode

El alcance del entrenamiento que se realizará al crear la versión de la solución. Un entrenamiento `FULL` tiene en cuenta todos los datos del grupo de conjuntos de datos. Con `UPDATE` se procesan solo los datos que han cambiado desde el último entrenamiento. Solo las versiones de solución creadas con la receta User-Personalization pueden usar `UPDATE`.

Tipo: cadena

Valores válidos: `FULL | UPDATE | AUTOTRAIN`

Obligatorio: no

trainingType

Indica si la versión de solución se creó de forma automática o manual.

Tipo: cadena

Valores válidos: `AUTOMATIC | MANUAL`

Obligatorio: no

tunedHPOParams

Si se realizó la optimización de hiperparámetros, contiene los valores de hiperparámetros del modelo con mejor rendimiento.

Tipo: objeto TunedHPOParams

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulte lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

SolutionVersionSummary

Servicio: Amazon Personalize

Proporciona un resumen de las propiedades de una versión de solución. Para obtener una lista completa, llame a la API DescribeSolutionVersion.

Contenido

creationDateTime

La fecha y hora (en tiempo Unix) en que creó esta versión de una solución.

Tipo: marca temporal

Obligatorio: no

failureReason

Si se produce un error en una versión de solución, indica el motivo.

Tipo: cadena

Requerido: no

lastUpdatedDateTime

La fecha y hora (en tiempo Unix) en que se actualizó por última vez la versión de solución.

Tipo: marca temporal

Obligatorio: no

solutionVersionArn

El nombre de recurso de Amazon (ARN) de la versión de solución.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

status

El estado de la versión de solución.

Una versión de la solución puede tener uno de los siguientes estados:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -o- CREATE FAILED

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

trainingMode

El alcance del entrenamiento que se realizará al crear la versión de la solución. Un entrenamiento `FULL` tiene en cuenta todos los datos del grupo de conjuntos de datos. Con `UPDATE` se procesan solo los datos que han cambiado desde el último entrenamiento. Solo las versiones de solución creadas con la receta User-Personalization pueden usar `UPDATE`.

Tipo: cadena

Valores válidos: `FULL | UPDATE | AUTOTRAIN`

Obligatorio: no

trainingType

Indica si la versión de solución se creó de forma automática o manual.

Tipo: cadena

Valores válidos: `AUTOMATIC | MANUAL`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Tag

Servicio: Amazon Personalize

Los metadatos opcionales que se aplican a los recursos para ayudarle a categorizarlos y organizarlos. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Para obtener más información, consulte Etiquetado de recursos de Amazon Personalize.

Contenido

tagKey

Una parte de un par clave-valor que compone una etiqueta. Un clave es una etiqueta general que actúa como una categoría para valores de etiqueta más específicos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

tagValue

La parte opcional de un par clave-valor que compone una etiqueta. Un valor actúa como un descriptor en una categoría de etiquetas (clave).

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

ThemeGenerationConfig

Servicio: Amazon Personalize

Detalles de la configuración para generar temas con un trabajo de inferencia por lotes.

Contenido

fieldsForThemeGeneration

Campos utilizados para generar temas descriptivos para un trabajo de inferencia por lotes.

Tipo: objeto FieldsForThemeGeneration

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

TrainingDataConfig

Servicio: Amazon Personalize

La configuración de los datos de entrenamiento que se utilizará al crear un recomendador de dominios o una versión de solución personalizada (modelo entrenado).

Contenido

excludedDatasetColumns

Especifica las columnas que se van a excluir del entrenamiento. Cada clave es un tipo de conjunto de datos y cada valor es una lista de columnas. Excluya columnas para controlar qué datos utiliza Amazon Personalize para generar recomendaciones.

Por ejemplo, puede tener una columna que desee utilizar únicamente para filtrar recomendaciones. Puede excluir esta columna del entrenamiento y Amazon Personalize solo la tendrá en cuenta al filtrar.

Tipo: mapa de cadena a matriz de cadenas

Entradas de mapa: número máximo de 3 elementos.

Limitaciones de longitud de clave: longitud máxima de 256.

Patrón de clave: `^[A-Za-z_]+$`

Miembros de la matriz: número máximo de 50 elementos.

Limitaciones de longitud: longitud máxima de 150 caracteres.

Patrón: `[A-Za-z_][A-Za-z\d_]*`

Obligatorio: no

includedDatasetColumns

Especifica las columnas que se van a incluir de la formación. Cada clave es un tipo de conjunto de datos y cada valor es una lista de columnas. Incluye columnas para controlar qué datos utiliza Amazon Personalize para generar recomendaciones.

Por ejemplo, es posible que tengas varias columnas en el conjunto de datos de tus artículos, pero solo quieras usar el título y la descripción para el entrenamiento. Puede incluir solo estas dos

columnas para la formación y Amazon Personalize solo considerará las columnas incluidas para la formación.

Tipo: mapa de cadena a matriz de cadenas

Entradas de mapa: número máximo de 3 elementos.

Limitaciones de longitud de clave: longitud máxima de 256.

Patrón de clave: `^[A-Za-z_]+$`

Miembros de la matriz: número máximo de 50 elementos.

Limitaciones de longitud: longitud máxima de 150 caracteres.

Patrón: `[A-Za-z_][A-Za-z\d_]*`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWSSDKs, consulte lo siguiente:

- AWSSDK para C++
- AWSSDK para Java V2
- AWSSDK para Ruby V3

TunedHPOParams

Servicio: Amazon Personalize

Si se realizó la optimización de hiperparámetros (HPO), contiene los valores de hiperparámetros del modelo con mejor rendimiento.

Contenido

algorithmHyperParameters

Una lista de los valores de hiperparámetros del modelo con mejor rendimiento.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 100 elementos.

Limitaciones de longitud de clave: longitud máxima de 256 caracteres.

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Eventos de Amazon Personalize

Eventos de Amazon Personalize admite los siguientes tipos de datos:

- Action
- ActionInteraction
- Event
- Item

- MetricAttribution
- User

Action

Servicio: Amazon Personalize Events

Representa los metadatos de las acciones agregadas a un conjunto de datos de acciones mediante la API `PutActions`. Para obtener más información, consulte Importación de acciones de forma individual.

Contenido

actionId

ID asociado con la acción.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

properties

Mapa de cadenas de metadatos específicos de la acción. Los elementos del mapa se componen de un par clave-valor. Por ejemplo, `{"value": "100"}`.

Las claves utilizan nombres con mayúsculas y minúsculas que coinciden con los campos del esquema del conjunto de datos de acciones. En el ejemplo anterior, `value` coincide con el campo “VALUE” definido en el esquema de acciones. En el caso de los datos de cadena categóricos, para incluir varias categorías para una sola acción, separe cada categoría con un separador vertical (`|`). Por ejemplo, `\"Deluxe|Premium\"`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 32 000.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

ActionInteraction

Servicio: Amazon Personalize Events

Representa un evento de interacción de acciones enviado mediante la API `PutActionInteractions`.

Contenido

actionId

ID de la acción con la que el usuario ha interactuado. Corresponde al campo `ACTION_ID` del esquema de interacción de acciones.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

eventType

Tipo de evento de interacción de acciones. Puede especificar los tipos de evento `Viewed`, `Taken` y `Not Taken`. Para obtener más información sobre los datos de los tipos de eventos de interacción de acciones, consulte Datos de tipo de eventos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

sessionId

ID asociado a la visita del usuario. La aplicación genera un `sessionId` único cuando un usuario visita por primera vez su sitio web o utiliza su aplicación.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

timestamp

Marca temporal de cuándo se produjo el evento de interacción de acciones. Las marcas temporales deben estar en formato de tiempo Unix expresado en segundos.

Tipo: marca temporal

Obligatorio: sí

eventId

ID asociado al evento. Si no se proporciona un ID de evento, Amazon Personalize genera un ID único para el evento. El ID de evento no se utiliza como entrada en el modelo. Amazon Personalize usa el ID del evento para distinguir eventos únicos. Los eventos posteriores al primero con el mismo ID de evento no se utilizan en el entrenamiento con modelos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

impression

Lista de ID de acciones que representa la secuencia de acciones que ha mostrado al usuario. Por ejemplo, `["actionId1", "actionId2", "actionId3"]`. Amazon Personalize no usa datos de impresiones de los eventos de interacción de acciones. En su lugar, registre varios eventos para cada acción y use el tipo de evento `Viewed`.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 25 elementos.

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

properties

Un mapa de cadenas de datos específicos de un evento que puede decidir registrar. Por ejemplo, si un usuario realiza una acción, aparte del ID de la acción, también puede enviar el número de acciones realizadas por el usuario.

Los elementos del mapa se componen de un par clave-valor. Por ejemplo:

```
{"numberOfActions": "12"}
```

Las claves utilizan nombres con mayúsculas y minúsculas que coinciden con los campos del esquema de interacciones de acción. En el ejemplo anterior, `numberOfActions` coincidiría con el campo “NUMBER_OF_ACTIONS” definido en el esquema de interacciones de acción.

No se puede incluir lo siguiente como palabra clave para las propiedades (no distingue entre mayúsculas y minúsculas).

- userId
- sessionId
- eventType
- marca de tiempo
- recommendationId
- impression

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Obligatorio: no

recommendationId

ID de la lista de recomendaciones que contiene la acción con la que ha interactuado el usuario.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 40 caracteres.

Obligatorio: no

userId

ID del usuario que ha interactuado con la acción. Corresponde al campo `USER_ID` del esquema de interacciones de acción.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Event

Servicio: Amazon Personalize Events

Representa la información de eventos de interacción de elementos enviada mediante la API `PutEvents`.

Contenido

eventType

El tipo de evento, como hacer clic o descargar. Esta propiedad corresponde al campo `EVENT_TYPE` de su esquema del conjunto de datos de interacciones de elementos y depende de los tipos de eventos que esté rastreando.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

sentAt

La marca de tiempo (en tiempo Unix) del lado del cliente cuando ocurrió el evento.

Tipo: marca temporal

Obligatorio: sí

eventId

ID asociado al evento. Si no se proporciona un ID de evento, Amazon Personalize genera un ID único para el evento. El ID de evento no se utiliza como entrada en el modelo. Amazon Personalize usa el ID del evento para distinguir eventos únicos. Los eventos posteriores al primero con el mismo ID de evento no se utilizan en el entrenamiento con modelos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

eventValue

Valor del evento que corresponde al campo `EVENT_VALUE` del esquema de interacciones de elementos.

Tipo: flotante

Obligatorio: no

impression

Una lista de ID de elementos que representa la secuencia de elementos que ha mostrado al usuario. Por ejemplo, `["itemId1", "itemId2", "itemId3"]`. Proporcione una lista de elementos para registrar manualmente los datos de las impresiones de un evento. Para obtener más información sobre el registro de los datos de impresiones, consulte Registro de los datos de impresiones.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 25 elementos.

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

itemId

Clave del ID de elemento que corresponde al campo `ITEM_ID` del esquema del conjunto de datos de interacciones de elementos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

metricAttribution

Contiene información sobre la atribución de métricas asociada a un evento. Para obtener más información sobre la atribución de métricas, consulte Medición del impacto de las recomendaciones.

Tipo: objeto MetricAttribution

Obligatorio: no

properties

Un mapa de cadenas de datos específicos de un evento que puede decidir registrar. Por ejemplo, si un usuario valora una película en su sitio, aparte del ID de película (`itemId`) y la valoración

(`eventValue`), también puede enviar el número de valoraciones de películas realizadas por el usuario.

Los elementos del mapa se componen de un par clave-valor. Por ejemplo:

```
{"numberOfRatings": "12"}
```

Las claves utilizan nombres con mayúsculas y minúsculas que coinciden con los campos del esquema del conjunto de datos de interacciones de elementos. En el ejemplo anterior, `numberOfRatings` coincidiría con el campo “NUMBER_OF_RATINGS” definido en el esquema del conjunto de datos de interacciones de elementos.

No se puede incluir lo siguiente como palabra clave para las propiedades (no distingue entre mayúsculas y minúsculas).

- userId
- sessionId
- eventType
- marca de tiempo
- recommendationId
- impression

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Obligatorio: no

recommendationId

El ID de la lista de recomendaciones que contiene el elemento con el que interactuó el usuario. Proporcione un `recommendationId` para que Amazon Personalize registre implícitamente las recomendaciones que muestra a su usuario como datos de impresiones. O proporcione `recommendationId` si utiliza una atribución de métricas para medir el impacto de las recomendaciones.

Para obtener más información sobre el registro de los datos de impresiones, consulte Registro de los datos de impresiones. Para obtener más información sobre la creación de una atribución de métricas, consulte Medición del impacto de las recomendaciones.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 40 caracteres.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Item

Servicio: Amazon Personalize Events

Representa los metadatos de los elementos agregados a un conjunto de datos de elementos mediante la API `PutItems`. Para obtener más información, consulte Importación de elementos de forma individual.

Contenido

itemId

El ID asociado al elemento.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

properties

Un mapa de cadenas de metadatos específicos de un elemento. Los elementos del mapa se componen de un par clave-valor. Por ejemplo, `{"numberOfRatings": "12"}`.

Las claves utilizan nombres con mayúsculas y minúsculas que coinciden con los campos del esquema del conjunto de datos de elementos. En el ejemplo anterior, `numberOfRatings` coincide con el campo “NUMBER_OF_RATINGS” definido en el esquema de elementos. En el caso de los datos de cadena categóricos, para incluir varias categorías para un solo elemento, separe cada categoría con un separador vertical (`|`). Por ejemplo, `\"Horror|Action\"`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 32 000.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

MetricAttribution

Servicio: Amazon Personalize Events

Contiene información sobre una atribución de métricas asociada a un evento. Para obtener más información sobre la atribución de métricas, consulte Medición del impacto de las recomendaciones.

Contenido

eventAttributionSource

El origen del evento, por ejemplo, un tercero.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 1024 caracteres.

Patrón: `^[\x20-\x7E]*[\x21-\x7E]+[\x20-\x7E]*$`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

User

Servicio: Amazon Personalize Events

Representa los metadatos de usuario agregados a un conjunto de datos de usuarios mediante la API `PutUsers`. Para obtener más información, consulte Importación de usuarios de forma individual.

Contenido

userId

El ID asociado al usuario.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: sí

properties

Un mapa de cadenas de metadatos específicos del usuario. Los elementos del mapa se componen de un par clave-valor. Por ejemplo, `{"numberOfVideosWatched": "45"}`.

Las claves utilizan nombres con mayúsculas y minúsculas que coinciden con los campos del esquema del conjunto de datos de usuarios. En el ejemplo anterior, `numberOfVideosWatched` coincidiría con el campo “NUMBER_OF_VIDEOS_WATCHED” definido en el esquema de usuarios. En el caso de los datos de cadena categóricos, para incluir varias categorías para un solo usuario, separe cada categoría con un separador vertical (`|`). Por ejemplo, `\"Member|Frequent shopper\"`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 24 000.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++

- AWS SDK para Java V2
- AWS SDK para Ruby V3

Versión ejecutable de Amazon Personalize

Versión ejecutable de Amazon Personalize admite los siguientes tipos de datos:

- PredictedAction
- PredictedItem
- Promotion

PredictedAction

Servicio: Amazon Personalize Runtime

Objeto que identifica una acción.

La API GetActionRecommendations devuelve una lista de instancias de `PredictedAction`.

Contenido

actionId

ID de la acción recomendada.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

score

Puntuación de la acción recomendada. Para obtener información sobre las puntuaciones de las acciones, consulte Cómo funciona la puntuación de recomendaciones de acciones.

Tipo: Doble

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

PredictedItem

Servicio: Amazon Personalize Runtime

Objeto que identifica un elemento.

Las API GetRecommendations y GetPersonalizedRanking y devuelven una lista de `PredictedItem`.

Contenido

itemId

El ID del elemento recomendado.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Obligatorio: no

metadata

Metadatos sobre el elemento del conjunto de datos de elementos.

Tipo: mapa de cadena a cadena

Limitaciones de longitud de clave: longitud máxima de 150 caracteres.

Limitaciones de longitud de valor: longitud máxima de 20 000.

Obligatorio: no

promotionName

El nombre de la promoción que incluía el elemento previsto.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

reason

Si utiliza User-Personalization-v2, una lista de los motivos por los que se ha incluido el elemento en las recomendaciones. Algunos motivos posibles son los siguientes:

- Elemento promocionado: indica que el elemento se ha incluido como parte de una promoción que ha aplicado en la solicitud de recomendación.
- Exploración: indica que el elemento se ha incluido con la exploración. Con la exploración, las recomendaciones incluyen elementos con menos datos de interacciones o relevancia para el usuario. Para obtener más información sobre la exploración, consulte Exploración.
- Elemento popular: indica que el elemento se ha incluido como marcador de posición de elemento popular. Si utiliza un filtro, en función del número de recomendaciones que elimina el filtro, Amazon Personalize podría agregar marcadores de posición de elemento para `numResults` correspondiente a la solicitud de recomendación. Estos elementos son populares, según los datos de interacciones, que cumplen los criterios de filtro. No tienen una puntuación de relevancia para el usuario.

Tipo: matriz de cadenas

Limitaciones de longitud: longitud máxima de 256 caracteres.

Obligatorio: no

score

Representación numérica de la certeza del modelo de que el elemento será la siguiente selección del usuario. Para obtener más información sobre la lógica de puntuación, consulte Puntuaciones de recomendaciones.

Tipo: Doble

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

Promotion

Servicio: Amazon Personalize Runtime

Contiene información sobre una promoción. Una promoción define reglas empresariales adicionales que se aplican a un subconjunto configurable de elementos recomendados.

Contenido

filterArn

El nombre de recurso de Amazon (ARN) del filtro que utiliza la promoción. Este filtro define los criterios de los artículos promocionados. Para obtener más información, consulte Filtros de promoción.

Tipo: cadena

Limitaciones de longitud: longitud máxima de 256.

Patrón: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Obligatorio: no

filterValues

Los valores que se deben utilizar al promocionar elementos. Para cada parámetro de marcador de posición de la expresión de filtro de la promoción, proporcione el nombre del parámetro (respetando mayúsculas y minúsculas) como clave y los valores del filtro como valor correspondiente. Separe los valores múltiples de un parámetro con una coma.

En el caso de las expresiones de filtro que utilizan un elemento `INCLUDE` para incluir elementos, debe proporcionar valores para todos los parámetros definidos en la expresión. En el caso de los filtros con expresiones que utilizan un elemento `EXCLUDE` para excluir elementos, puede omitir `filter-values`. En este caso, Amazon Personalize no utiliza esa parte de la expresión para filtrar recomendaciones.

Para obtener más información sobre la creación de filtros, consulte Filtrado de recomendaciones y segmentos de usuario.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 25 elementos.

Limitaciones de longitud de clave: longitud máxima de 50 caracteres.

Patrón de clave: `[A-Za-z0-9_]+`

Limitaciones de longitud de valor: longitud máxima de 1000 caracteres.

Obligatorio: no

name

El nombre de la promoción.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 63.

Patrón: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Obligatorio: no

percentPromotedItems

El porcentaje de elementos recomendados a los que se debe aplicar la promoción.

Tipo: número entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un SDK de AWS de un idioma específico, consulte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Tipos de errores comunes

En esta sección se enumeran los tipos de errores más comunes que puede devolver este AWS servicio. No todos los servicios devuelven todos los tipos de errores que se indican aquí. En el caso

de los errores específicos de una acción de la API de este servicio, consulte el tema de dicha acción de la API.

AccessDeniedException

No tiene permiso para realizar esta acción. Compruebe que su política de IAM incluya los permisos necesarios.

Código de estado HTTP: 403

ExpiredTokenException

El token de seguridad incluido en la solicitud ha caducado. Solicita un nuevo token de seguridad e inténtalo de nuevo.

Código de estado HTTP: 403

IncompleteSignature

La firma de la solicitud no cumple con los AWS estándares. Comprueba que utilizas AWS credenciales válidas y que tu solicitud tiene el formato correcto. Si utilizas un SDK, asegúrate de que esté actualizado.

Código de estado HTTP: 403

InternalFailure

La solicitud no se puede procesar en este momento debido a un problema interno del servidor. Inténtelo de nuevo más tarde. Si el problema persiste, ponte en contacto con AWS Support.

Código de estado HTTP: 500

MalformedHttpRequestException

No se puede procesar el cuerpo de la solicitud. Esto suele ocurrir cuando el cuerpo de la solicitud no se puede descomprimir mediante el algoritmo de codificación de contenido especificado. Compruebe que el encabezado de codificación del contenido coincide con el formato de compresión utilizado.

Código de estado HTTP: 400

NotAuthorized

No tienes permisos para realizar esta acción. Compruebe que su política de IAM incluya los permisos necesarios.

Código de estado HTTP: 401

OptInRequired

Tu AWS cuenta necesita una suscripción a este servicio. Comprueba que has activado el servicio en tu cuenta.

Código de estado HTTP: 403

RequestAbortedException

La solicitud se canceló antes de que se pudiera devolver una respuesta. Esto suele ocurrir cuando el cliente cierra la conexión.

Código de estado HTTP: 400

RequestEntityTooLargeException

La entidad de solicitud es demasiado grande. Reduce el tamaño del cuerpo de la solicitud e inténtalo de nuevo.

Código de estado HTTP: 413

RequestTimeoutException

Se agotó el tiempo de espera de la solicitud. El servidor no recibió la solicitud completa en el plazo previsto. Inténtelo de nuevo.

Código de estado HTTP: 408

ServiceUnavailable

El servicio no está disponible temporalmente. Inténtelo de nuevo más tarde.

Código de estado HTTP: 503

ThrottlingException

Tu porcentaje de solicitudes es demasiado alto. Reintentan AWS SDKs automáticamente las solicitudes que reciben esta excepción. Reduzca la frecuencia de las solicitudes.

Código de estado HTTP: 400

UnknownOperationException

No se reconoce la acción o la operación. Comprueba que el nombre de la acción está escrito correctamente y que es compatible con la versión de API que utilizas.

Código de estado HTTP: 404

UnrecognizedClientException

El identificador de clave de AWS acceso o certificado X.509 que has proporcionado no existe en nuestros registros. Compruebe que utiliza credenciales válidas y que no hayan caducado.

Código de estado HTTP: 403

ValidationError

La entrada no cumple con el formato o las restricciones requeridos. Compruebe que se incluyen todos los parámetros necesarios y que los valores son válidos.

Código de estado HTTP: 400

Parámetros comunes

La siguiente lista contiene los parámetros que utilizan todas las acciones para firmar solicitudes de Signature Version 4 con una cadena de consulta. Los parámetros específicos de acción se enumeran en el tema correspondiente a la acción. Para obtener más información sobre la versión 4 de Signature, consulte Firmar solicitudes de AWS API en la Guía del usuario de IAM.

X-Amz-Algorithm

El algoritmo de hash que utilizó para crear la solicitud de firma.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Valores válidos: `AWS4-HMAC-SHA256`

Obligatorio: condicional

X-Amz-Credential

El valor del ámbito de la credencial, que es una cadena que incluye la clave de acceso, la fecha, la región a la que se dirige, el servicio que solicita y una cadena de terminación (“aws4_request”). El valor se expresa en el siguiente formato: access_key/AAAAMMDD/region/service/aws4_request.

Para obtener más información, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-Date

La fecha utilizada para crear la firma. El formato debe ser ISO 8601 formato básico (AAAAMMDD'T'HHMMSS'Z'). Por ejemplo, la siguiente fecha y hora es un X-Amz-Date valor válido:`20120325T120000Z`.

Condición: X-Amz-Date es opcional para todas las solicitudes; se puede usar para anular la fecha utilizada para firmar las solicitudes. Si el encabezado de fecha se especifica en el formato básico ISO 8601, no X-Amz-Date es obligatorio. Cuando X-Amz-Date se usa, siempre anula el valor del encabezado de fecha. Para obtener más información, consulte Elementos de la firma de una solicitud de AWS API en la Guía del usuario de IAM.

Tipo: cadena

Obligatorio: condicional

X-Amz-Security-Token

El token de seguridad temporal que se obtuvo mediante una llamada a AWS Security Token Service (AWS STS). Para obtener una lista de servicios compatibles con las credenciales de seguridad temporales de AWS STS, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Condición: si utilizas credenciales de seguridad temporales de AWS STS, debes incluir el token de seguridad.

Tipo: cadena

Obligatorio: condicional

X-Amz-Signature

Especifica la firma codificada hexadecimal que se calculó a partir de la cadena que se va a firmar y la clave de firma derivada.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-SignedHeaders

Especifica todos los encabezados HTTP que se incluyeron como parte de la solicitud canónica. Para obtener más información sobre cómo especificar encabezados firmados, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

Historial de documentos para Amazon Personalize

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía para desarrolladores de Amazon Personalize. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Nueva característica	Amazon Personalize presenta la fórmula de similitud semántica que define la similitud de los elementos en función de metadatos, como los títulos y las descripciones, con una clasificación opcional por popularidad o marca temporal, soporte de formación incremental y selección flexible de columnas para la formación. Para obtener más información, consulte la receta de similitud semántica.	30 de noviembre de 2025
Nueva característica	Amazon Personalize ahora admite la configuración de soluciones para asignar diferentes ponderaciones a los distintos tipos de eventos. Para obtener más información, consulte Optimización de una solución con la configuración de eventos.	11 de abril de 2025
Nueva característica	Amazon Personalize ahora admite IPv6 el tráfico mediante puntos de enlace de doble	31 de enero de 2025

	pila. Para ver una lista de los puntos de conexión de Amazon Personalize por región, consulte Regiones y puntos de conexión de AWS en la Referencia general de Amazon Web Services.	
Nueva característica de documentación	Con los datos de páginas vistas y el análisis en profundidad, la estructura de la guía para desarrolladores de Amazon Personalize se ha actualizado para aumentar la visibilidad de los temas importantes. Se ha actualizado la navegación para reducir la profundidad total y se han consolidado los temas relacionados. Por ejemplo, ahora puede encontrar toda la guía de preparación de datos en el tema Preparación de los datos de entrenamiento para Amazon Personalize. Para ver un tema que lo ayude a navegar por la nueva estructura, consulte Detalles del flujo de trabajo de Amazon Personalize. En esta página, se asigna cada paso del flujo de trabajo de Amazon Personalize al tema asociado de la guía.	3 de septiembre de 2024

Nueva característica	Ahora puede actualizar una solución de Amazon Personalize para cambiar la configuración de entrenamiento. Puede activar o desactivar el entrenamiento automático y cambiar la frecuencia del entrenamiento. Para obtener más información, consulte Actualización de una solución para cambiar la configuración de entrenamiento automático.	29 de agosto de 2024
Nueva característica de documentación	El tema Cómo personalizar los resultados de búsqueda a partir del OpenSearch tema de la guía para desarrolladores de Amazon Personalize se ha reorganizado por flujos de trabajo de código OpenSearch abierto y de OpenSearch servicio. Ahora incluye nuevos subtemas: Personalización de los resultados de Amazon OpenSearch Service con Amazon Personalize y Personalización de los resultados de Open Search de código abierto con Amazon Personalize.	23 de agosto de 2024

Nueva característica	Amazon Personalize ahora admite la posibilidad de eliminar usuarios y los datos, incluidos los metadatos y los datos de interacciones, de un grupo de conjuntos de datos. Para obtener más informaci ón, consulte Eliminación de usuarios.	2 de mayo de 2024
Nueva característica	Amazon Personalize ahora puede tener en cuenta hasta cinco millones de elementos con un entrenamiento más rápido, una latencia de recomendación más baja y recomendaciones más relevantes. Dos nuevas recetas personalizadas usan este entrenamiento mejorado, User-Personalization-v2 y Personalized-Ranking-v2.	1 de mayo de 2024
Nueva característica	Todas las nuevas solucione s personalizadas de Amazon Personalize ahora utilizan el entrenamiento automático. Para obtener más informaci ón sobre la configuración del entrenamiento automático, consulte Configuración del entrenamiento automático.	19 de abril de 2024

Nueva característica	Ahora, Amazon Personalize permite recomendar la siguiente mejor acción para un usuario en función de su comportamiento. Para obtener más información, consulta Next-Best-Action la receta.	26 de noviembre de 2023
Nueva característica	Ahora, Amazon Personalize permite incluir temas descriptivos en las recomendaciones por lotes con ayuda de la IA generativa. Para obtener información sobre cómo generar recomendaciones por lotes con temas, consulte Recomendaciones por lotes con temas. Para obtener información sobre el uso de Amazon Personalize con la IA generativa, consulte Amazon Personalize y la IA generativa.	26 de noviembre de 2023

Nueva característica	Ahora, Amazon Personalize permite incluir metadatos de un conjunto de datos de elementos en las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para una campaña, consulte Metadatos de elemento en las recomendaciones. Para obtener información sobre cómo habilitar los metadatos para un recomendador, consulte Habilitación de los metadatos en las recomendaciones para un recomendador.	26 de noviembre de 2023
Nueva característica	Amazon Personalize ahora admite la posibilidad de personalizar los resultados de búsqueda del OpenSearch Servicio. Para obtener más información, consulte Personalización de los resultados de búsqueda desde OpenSearch.	16 de octubre de 2023

Nueva característica	Amazon Personalize ahora admite la posibilidad de importar hasta 100 columnas de metadatos para conjuntos de datos de elementos (en lugar de 50) y 25 columnas de metadatos para conjuntos de datos de usuarios (en lugar de 5). Para obtener más información sobre los límites de Amazon Personalize, consulte Puntos de conexión y cuotas de Amazon Personalize.	5 de septiembre de 2023
Nueva característica	Amazon Personalize ahora admite la posibilidad de personalizar los resultados de búsqueda desde OpenSearch. Para obtener más información, consulte Personalización de los resultados de búsqueda desde OpenSearch (autogestionados).	25 de julio de 2023
Nueva característica	Amazon Personalize ahora admite la posibilidad de sustituir el esquema de un conjunto de datos por uno nuevo o existente. Para obtener más información, consulte Sustitución del esquema de un conjunto de datos.	13 de julio de 2023

Nueva característica	Si utiliza las recetas User-Personalization o Personalized-Ranking, los trabajos de inferencia por lotes de Amazon Personalize ahora pueden utilizar los datos que importe de manera incremental sin necesidad de reentrenamiento. Para obtener más información, consulte Obtención de recomendaciones por lotes.	30 de junio de 2023
Nueva característica	Amazon Personalize ahora admite la posibilidad de filtrar elementos en función del elemento que especifique en su solicitud de recomendaciones de elementos relacionados. Para obtener más información sobre los filtros, consulte Filtrado de recomendaciones y segmentos de usuario.	21 de junio de 2023
Nueva característica	Amazon Personalize ahora admite la posibilidad de filtrar elementos en función del elemento que especifique en su solicitud de recomendaciones de elementos relacionados. Para obtener más información sobre los filtros, consulte Filtrado de recomendaciones y segmentos de usuario.	21 de junio de 2023

Nueva característica	Amazon Personalize ahora admite conexiones privadas entre una nube virtual privada (VPC) y Amazon Personalize con un punto de conexión de VPC de Amazon de interfaz. Para obtener más información, consulte Amazon Personalize y puntos de conexión de la VPC de tipo interfaz (AWS PrivateLink).	12 de junio de 2023
Nueva característica	Amazon Personalize ahora admite la configuración de las columnas que se utilizan al entrenar cuando crea un generador de recomendaciones o una solución personalizada. Para obtener información sobre cómo configurar las columnas al crear un generador de recomendaciones, consulte Creación de generadores de recomendaciones. Para obtener información sobre cómo configurar columnas al crear una solución, consulte Configuración de las columnas que se utilizan en el entrenamiento.	30 de mayo de 2023

Nueva característica de documentación	La guía para desarrolladores de Amazon Personalize ahora incluye información sobre cómo realizar A/B pruebas con las recomendaciones de Amazon Personalize. Para obtener más información, consulte Medir el impacto de las recomendaciones con A/B las pruebas.	5 de mayo de 2023
Nueva característica	Amazon Personalize ahora admite la configuración del modo en que la popularid ad influye en las recomenda ciones generadas por la receta Similar-Items. Para obtener más información, consulte Receta de elementos similares .	21 de abril de 2023
Nueva característica	Amazon Personalize ahora admite el uso de Amazon SageMaker AI Data Wrangler para importar datos de más de 40 fuentes a conjuntos de datos de Amazon Personalize. Para obtener más información, consulte Importación de datos con Amazon SageMaker AI Data Wrangler.	14 de abril de 2023

Nueva característica de documentación	La guía para desarrolladores de Amazon Personalize ahora incluye una nueva lista de verificación de preparación. Esta lista de verificación le ayuda a prepararse para utilizar Amazon Personalize con sus propios datos. Para obtener más información, consulte Lista de verificación de preparación.	9 de febrero de 2023
Nueva característica	Amazon Personalize ahora permite generar información y estadísticas para los datos que importe en los conjuntos de datos. Para obtener más información, consulte Análisis de datos en conjuntos de datos.	25 de enero de 2023
Nueva característica	Amazon Personalize ahora admite una nueva receta Trending-Now para grupos de conjuntos de datos personalizados. Para obtener más información, consulte Receta Trending-Now.	6 de enero de 2023
Nueva característica	Amazon Personalize ahora admite el uso de etiquetas en las políticas de IAM. Para obtener más información, consulte Uso de etiquetas en políticas de IAM.	28 de diciembre de 2022

Nueva característica	Amazon Personalize ahora permite realizar ajustes en la cuota Número máximo de interacciones que se consideran en un modelo durante el entrenamiento. Además, ya no se aplica la cuota Número máximo de usuarios que se considera n en un modelo durante el entrenamiento. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon Personalize.	15 de diciembre de 2022
Nueva característica	Amazon Personalize ahora admite la creación de una atribución métrica para medir el impacto empresarial de las recomendaciones. Para obtener más información, consulte Medición del impacto de las recomendaciones.	17 de noviembre de 2022

Nueva característica	Ahora se han incrementado las cuotas de Amazon Personalize para el número total de soluciones, campañas activas, generadores de recomendaciones y filtros. Ahora, cada una de estas cuotas se aplica por grupo de conjuntos de datos en lugar de por cuenta. Para obtener información sobre cuotas, consulte Cuotas y puntos de conexión de Amazon Personalize.	7 de septiembre de 2022
Nueva característica	Los filtros de Amazon Personalize ahora tienen en cuenta hasta 100 interacciones por usuario y por tipo de evento. Para obtener más información sobre los filtros, consulte Filtrado de recomendaciones y segmentos de usuario.	29 de agosto de 2022
Nueva característica	Amazon Personalize ahora admite un nuevo caso de uso de Trending-Now para el dominio de VIDEO_ON_DEMAND. Para obtener más información, consulte Casos de uso de VIDEO_ON_DEMAND.	17 de agosto de 2022

Nueva característica	Amazon Personalize ahora admite la promoción de elementos en recomenda ciones con un filtro de promoción independiente. Para obtener informaci ón sobre la promoción de elementos, consulte Promoción de elementos en las recomendaciones.	12 de agosto de 2022
Nueva característica	Amazon Personalize ahora admite el uso de operadores de comparación en expresion es de filtro con parámetros de marcador de posición. Para obtener información sobre las expresiones de filtro, consulte Expresiones de filtro.	12 de agosto de 2022
Nueva característica	Amazon Personalize ahora admite actualizaciones masivas incrementales de conjuntos de datos. Ahora puede usar un trabajo de importación de conjuntos de datos para actualizar un conjunto de datos sin sustituir los datos existentes. Para obtener más información, consulte Actualización de datos masivos existentes.	2 de agosto de 2022

Nueva característica	Amazon Personalize ahora puede usar metadatos de texto no estructurado en distintos idiomas. Para obtener más información, consulte Metadatos de texto no estructurado.	6 de junio de 2022
Nueva característica	Los generadores de recomendaciones de Amazon Personalize ahora generan métricas sin conexión. Puede utilizar estas métricas para evaluar el rendimiento de su generador de recomenda ciones. Para obtener más información, consulte Evaluación de un recomenda dor.	24 de mayo de 2022
Nueva característica	Amazon Personalize ahora admite la posibilidad de detener un generador de recomendaciones y de reiniciarlo más adelante. De esta forma, puede pausar la facturación del generador de recomendaciones y pagarlo solo cuando lo utilice. Para obtener más información, consulte Detención e inicio de un generador de recomenda ciones.	20 de abril de 2022

Nueva característica	Amazon Personalize ahora admite el uso de etiquetas para clasificar y gestionar los recursos de Amazon Personalize. Para obtener más informaci ón, consulte Etiquetado de recursos de Amazon Personali ze.	7 de abril de 2022
Nueva característica	Amazon Personalize ahora admite la especificación de recursos conCloudFormation. Para obtener más información, consulte Especificar recursos con CloudFormation.	11 de marzo de 2022
Nueva característica de documentación	La guía para desarrolladores de Amazon Personalize ahora incluye un nuevo tema de solución de problemas que proporciona respuesta s a preguntas habituales y consejos para los mensajes de error que puede encontrar con Amazon Personalize. Para más información, consulte Solución de problemas.	15 de febrero de 2022

Nueva característica	Amazon Personalize ahora admite la creación de un grupo de conjuntos de datos de dominios con recursos optimizados de casos de uso para dominios de vídeo en diferido o de comercio electrónico. Para obtener más información, consulte Grupos de conjuntos de datos de dominio.	29 de noviembre de 2021
Nueva característica	Amazon Personalize ahora admite la creación de segmentos de usuarios con las nuevas recetas USER_SEGMENTATION. Las recetas USER_SEGM ENTATION generan segmentos de usuarios en función de los datos de entrada de los elementos. Para obtener más información, consulta la receta y Item-Attr ibute-Affinity la receta de Item-Afinidad.	29 de noviembre de 2021

Nueva característica	Amazon Personalize ahora admite una nueva receta de recomendación RELATED_ITEMS Similar-Items. Use la receta Similar-Items para generar recomendaciones para elementos similares en función tanto de los datos de interacciones como de los metadatos de elementos. Para obtener más información, consulte Receta de elementos similares.	5 de octubre de 2021
Nueva característica de documentación	La guía para desarrolladores de Amazon Personalize ahora incluye un tutorial de introducción al uso de Amazon Personalize con el SDK para Java 2.x. Para obtener más información, consulte Introducción (SDK para Java 2.x).	25 de agosto de 2021
Nueva característica	Amazon Personalize ahora puede extraer información significativa de los metadatos de texto no estructurado de un conjunto de datos de elementos. Para obtener más información, consulte Conjunto de datos de elementos.	9 de junio de 2021

Nueva característica	Amazon Personalize ahora admite la capacidad de dejar de crear una versión de la solución (dejar de entrenar un modelo). Para obtener más información, consulte Detención de la creación de una versión de solución.	20 de mayo de 2021
Nueva característica (versión preliminar)	Amazon Personalize ahora puede optimizar una solución para un objetivo, además de maximizar la relevancia, como maximizar los ingresos. Esta característica se encuentra en una versión preliminar. Para obtener más información, consulte Optimización de una solución para un objetivo adicional.	18 de mayo de 2021
Nueva característica	Amazon Personalize ahora puede exportar los registros de un conjunto de datos de Amazon Personalize en un bucket de Amazon S3 para su análisis y seguimiento. Para obtener más información, consulte Exportación de un conjunto de datos.	26 de abril de 2021

Nueva característica	Amazon Personalize ahora actualiza automáticamente el último modelo (versión de la solución) que haya entrenado con User-Personalization cada dos horas para incluir nuevos datos. Para obtener más información, consulte Receta User-personalization.	17 de noviembre de 2020
Nueva característica	Amazon Personalize ahora puede filtrar las recomenda ciones en función de los criterios que especifique al recibir recomendaciones. Para obtener más informaci ón, consulte Filtrado de recomendaciones.	10 de noviembre de 2020
Nueva característica	Amazon Personalize ahora admite la capacidad de importar usuarios y elementos de forma incremental. Para obtener más informaci ón, consulte Importación incremental de registros.	2 de octubre de 2020

Nueva característica	Amazon Personalize ahora admite una nueva receta de recomendación USER_PERSONALIZATION. Las características de USER_PERSONALIZATION incluyen el modelado de los datos de impresión, la exploración automática de elementos y la selección automática de elementos en frío. Para obtener más información, consulte Receta User-personalization.	5 de agosto de 2020
Nueva característica	Amazon Personalize ahora puede filtrar las recomendaciones en función de los metadatos de los elementos y los usuarios mediante expresiones de filtro personalizadas. Para obtener más información, consulte Filtrado de recomendaciones.	31 de julio de 2020
Nueva característica	Amazon Personalize ahora le permite filtrar resultados en función de los elementos con los que ha interactuado un usuario. Para obtener más información, consulte Filtrado de recomendaciones.	3 de junio de 2020

Nueva característica	Amazon Personalize ahora expone las puntuaciones para los elementos recomendados. Las puntuaciones represent an la certeza del modelo de Amazon Personalize de que un usuario vaya a elegir a continuación un determina do elemento. Para obtener más información, consulte Obtención de recomenda ciones.	3 de abril de 2020
Nueva región	Amazon Personalize ahora es compatible con la región Asia-Pacífico (Seúl). Para ver una lista completa de las regiones de AWS admitidas por Amazon Personalize, consulte la Tabla de regiones de AWS o Regiones y puntos de conexión de AWS en la Referencia general de Amazon Web Services.	21 de enero de 2020
Nueva característica	Amazon Personalize ahora puede obtener recomenda ciones basadas en metadatos contextuales. Para obtener más información, consulte Obtención de recomenda ciones.	19 de diciembre de 2019

Nuevas regiones	Amazon Personalize añade soporte para las regiones de Asia-Pacífico (Bombay), Asia-Pacífico (Sídney) y Canadá (centro). Para obtener una lista completa de las AWS regiones compatibles con Amazon Personalize, consulte la AWStabla de AWSregiones o Regiones y puntos de enlace en la Referencia general de Amazon Web Services.	18 de diciembre de 2019
Nueva característica	Amazon Personalize ahora admite flujos de trabajo de recomendaciones por lotes. Para obtener más información, consulte Obtener recomendaciones por lotes.	14 de noviembre de 2019
Disponibilidad general de Amazon Personalize	Amazon Personalize está ahora disponible para uso general.	10 de junio de 2019
Versión preliminar de Amazon Personalize	Esta es la primera versión preliminar de la documentación de Amazon Personalize.	28 de noviembre de 2018

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.