aws

开发人员指南

Amazon Personalize



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

Amazon Personalize: 开发人员指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon Personalize？ 1
Amazon Personalize 的定价 2
面向首次使用的用户的指南 2
借助 Magic Movie Machine 了解 Amazon Personalize 2
浏览本指南中的入门材料 2
相关 AWS 服务和解决方案 3
Third-party 服务 3
了解详情 4
使用 AWS 软件开发工具包 4
Amazon Personalize 和生成式人工智能 6
内容生成器中带有主题的建议 6
建议元数据 7
预配置的用于个性化的 LangChain 代码 7
工作方式 9
Amazon Personalize 工作流程 9
Amazon Personalize 术语 10
数据导入和管理 10
训练 13
模型部署和建议 15
Amazon Personalize 数据 16
交互数据 16
物品数据 17
用户数据 17
操作数据 17
操作交互数据 17
设置 Amazon Personalize 19
注册获取 AWS 账户 19
区域和端点 19
设置权限 19
向 Amazon Personalize 授予访问资源的权限 20
向用户授予访问 Amazon Personalize 的权限 23
向 Amazon Personalize 授予访问 Amazon S3 资源的权限 25
授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限 31
设置 AWS CLI 33

设置 AWS 软件开发工具包 34
入门教程 36
入门先决条件 36
创建训练数据（域数据集组） 37
创建训练数据（自定义数据集组） 38
域数据集组入门 39
域数据集组入门（控制台） 39
域数据集组入门 (SDK for Java 2.x) 45
域数据集组入门 (SDK for Python (Boto3)) 54
域数据集组入门 (SDK for JavaScript v3) 59
自定义数据集组入门 67
开始使用（控制台） 67
入门 (AWS CLI) 77
入门 (SDK for Python (Boto3)) 86
入门 (SDK for Java 2.x) 92
将自己的应用场景与 Amazon Personalize 资源进行匹配 104
使用案例和食谱特征 105
实时个性化 105
探索 105
自动更新 106
选择使用案例 108
VIDEO_ON_DEMAND 使用案例 108
ECOMMERCE 使用案例 112
选择食谱 116
按使用案例划分的 Amazon Personalize 食谱类型 117
Amazon Personalize 食谱 118
查看可用的 Amazon Personalize 食谱 119
User-Personalization-v2 120
User-Personalization 123
Trending-Now 136
热门程度-计数 139
Personalized-Ranking-v2 139
Personalized-Ranking 142
语义相似度 147
Similar-Items 150
SIMS 153

Next-Best-Action 159
Item-Affinity 164
Item-Attribute-Affinity 165
旧版配方 167
准备训练数据 183
适用于所有数据类型的批量数据格式准则 184
物品交互数据 185
物品交互数据要求 185
时间戳数据 186
事件类型和事件值数据 187
上下文元数据 188
展示数据 189
交互数据示例 192
物品元数据 193
物品数据要求 194
创建时间戳数据 196
分类元数据 196
非结构化文本元数据 197
数值数据 198
非分类字符串数据 198
物品元数据示例 198
用户元数据 200
用户数据要求 200
分类元数据 201
非分类字符串数据 201
用户元数据示例 201
操作元数据 202
操作数据要求 203
操作过期时间戳数据 204
重复频率数据 204
Value data 204
创建时间戳数据 205
分类元数据 205
非分类字符串数据 205
操作元数据示例 205
操作交互数据 207

操作交互数据要求 207
事件类型数据 208
操作交互数据示例 208
创建架构 JSON 文件 210
架构格式设置要求 210
架构数据类型 211
VIDEO_ON_DEMAND 数据集和架构 212
VIDEO_ON_DEMAND 域数据集和架构要求 212
物品交互数据集要求 213
用户数据集要求 215
物品数据集要求 216
ECOMMERCE 数据集和架构 219
ECOMMERCE 域数据集和架构要求 219
物品交互数据集要求 221
用户数据集要求 222
物品数据集要求 224
自定义数据集和架构 226
自定义数据集和架构要求 226
物品交互数据集架构要求 229
用户数据集架构要求 231
物品数据集架构要求 233
操作数据集架构要求 235
操作交互数据集架构要求 237
准备清单 239
您是否将自己的使用案例与 Amazon Personalize 资源进行了匹配？ 239
您是否有足够的物品交互数据？ 240
您是否有实时事件流式传输架构？ 240
您的数据是否针对 Amazon Personalize 进行了优化？ 240
您是否收集可以改进建议的可选数据？ 241
您是否有计划测试您的建议？ 241
您是否还有其他业务目标？ 241
创建数据集组 243
创建数据集组（控制台） 243
创建数据集组 (AWS CLI) 244
创建数据集组 (AWS SDK) 245
创建架构和数据集 248

创建数据集和架构（控制台） 248
创建数据集和架构 (AWS CLI) 249
创建数据集和架构 (AWS SDKs) 250
导入训练数据 255
导入批量数据 256
导入模式 257
创建数据集导入作业（控制台） 257
创建数据集导入作业（AWS CLI） 259
创建数据集导入作业（AWS SDKs） 261
使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据 265
附加信息 266
设置权限 266
从 Amazon Personalize 启动 Data Wrangler 268
将数据导入 Data Wrangler 269
转换数据 270
生成可视化和数据见解 271
处理数据并将其导入 Amazon Personalize 275
导入单个记录 277
单独导入交互 278
单独导入用户 280
单独导入物品 284
单独导入操作 289
分析训练数据 292
分析数据所需的权限 292
数据见解 293
查看数据集见解和统计信息 294
域推荐器 296
推荐器状态 297
每秒最低建议请求数和自动扩缩 297
创建域推荐器 298
创建推荐器（控制台） 298
创建推荐器（AWS CLI） 300
创建推荐器（AWS SDKs） 300
在建议中启用元数据 303
配置训练时使用的列 304
为域推荐器配置浏览 307

评估域推荐器 309
检索指标 310
指标定义 313
示例 314
其他资源 315
更新推荐器 315
更新推荐器（Amazon Personalize 控制台） 316
更新推荐器 (AWS CLI) 316
更新推荐器 (AWS SDK) 316
停止推荐器 318
停止推荐器（控制台） 319
停止推荐器（AWS CLI） 319
停止推荐器 (AWS SDK) 319
自定义资源 322
配置解决方案 323
创建解决方案 323
配置自动训练 330
配置训练时使用的列 334
针对其他目标优化解决方案 336
使用事件配置优化解决方案 342
超级参数和 HPO 347
选择用于训练的物品交互数据 351
克隆解决方案（控制台） 352
更新解决方案 353
更新解决方案（控制台） 354
更新解决方案 (AWS CLI) 354
更新解决方案 (AWS 软件开发工具包) 354
手动创建解决方案版本 355
创建解决方案版本（控制台） 356
创建解决方案版本 (AWS CLI) 357
创建解决方案版本 (AWS SDK) 358
停止创建解决方案版本 362
停止创建解决方案版本（控制台） 362
停止创建解决方案版本 (AWS CLI) 363
停止创建解决方案版本 (AWS SDKs) 363
评估解决方案版本 365

检索解决方案版本指标 366
指标定义 368
示例 373
其他资源 373
创建市场活动 374
市场活动自动更新 375
最低预调配的 TPS 375
推荐中的物品元数据 376
创建市场活动（控制台） 377
创建市场活动 (AWS CLI) 378
创建市场活动 (AWS SDK) 379
更新市场活动 382
更新市场活动（控制台） 383
更新广告活动 (AWS CLI) 383
更新广告活动 (AWS 软件开发工具包) 383
获取建议 386
建议分数 386
实时物品推荐 387
推荐评分的工作原理（自定义资源） 388
User-Personalization-v2 的推荐理由 388
获取实时物品推荐 389
获取推荐的物品元数据 393
推广物品 394
实时操作推荐 403
操作建议评分的工作原理 404
获取操作建议（控制台） 404
获取操作建议 (AWS CLI) 404
获取操作建议 (AWS SDK) 405
获取个性化排名（自定义资源） 405
个性化排名评分的工作原理 406
获取个性化排名（控制台） 406
获取个性化排名 (AWS CLI) 407
获取个性化排名 (AWS SDK) 408
示例笔记本 411
使用上下文元数据 412
使用情境相关元数据获取推荐 (AWSPython SDK) 412

获取批量物品推荐 413
批量工作流程 414
指南和要求 414
批量工作流程评分 414
带有主题的批量建议 415
为批量建议准备输入数据 417
创建批量推理作业 421
批量推理作业输出示例 430
获取批量用户细分 432
获取用户细分的准则和要求 433
为用户细分准备输入数据 433
创建批量细分作业 436
批量细分作业输出格式示例 442
筛选结果 444
筛选条件表达式 445
指南和要求 446
筛选条件表达式结构和元素 446
筛选器表达式示例 449
筛选实时建议 454
筛选实时建议（控制台） 454
筛选实时推荐 (AWS CLI) 460
筛选实时推荐 (AWS 软件开发工具包) 461
筛选批量建议和用户细分（自定义资源） 467
在您的输入 JSON 中提供筛选器值 467
筛选批量工作流程（控制台） 468
筛选批处理工作流程 (AWS 软件开发工具包) 468
记录事件 470
实时事件如何影响建议 470
记录物品交互事件 471
记录物品交互事件和训练模型的要求 472
创建物品交互事件跟踪器 472
记录单个物品交互事件 476
记录含事件值数据的多个物品交互事件 479
记录包含展示数据的物品交互事件 481
事件指标和归因报告 483
记录操作交互事件 485

记录操作交互事件的要求 486
操作交互事件跟踪器 ID 486
记录单个操作交互事件 486
记录多个操作交互事件 488
为匿名用户记录事件 489
为匿名用户构建持续的事件历史记录 490
第三方事件跟踪服务 490
示例实施 490
维护建议相关性 492
使数据集保持最新状态 492
维护域推荐器 492
维护自定义解决方案 493
训练后更新数据集中的数据 494
新数据如何影响实时建议 494
新的交互 495
新物品 495
新用户 496
新操作 497
新数据如何影响批量推荐（自定义资源） 497
新的交互 497
新用户 498
新物品 498
替换数据集的架构 499
指南和要求 499
替换数据集的架构（控制台） 500
替换数据集的架构 (AWS CLI) 500
替换数据集的架构 (AWS 软件开发工具包) 501
导出数据 502
数据集导出作业权限要求 503
用于导出数据集的服务角色策略 503
用于导出数据集的 Amazon S3 存储桶策略 503
在 Amazon Personalize 中创建数据集导出作业 504
创建数据集导出作业（控制台） 504
创建数据集导出任务 (AWS CLI) 505
创建数据集导出任务 (AWS 软件开发工具包) 507
删除资源 511

删除资源的准则 511
资源删除的推荐顺序 512
删除用户 513
指南和要求 513
准备待删除用户的列表 514
创建数据删除作业 515
删除数据集 518
删除数据集（控制台） 519
删除数据集 (AWS CLI) 519
删除数据集 (AWS SDKs) 519
衡量建议的影响 521
使用指标归因衡量建议的影响 521
指南和要求 522
创建指标归因 526
更新指标归因 532
删除指标归因 538
在中查看指标数据的图表 CloudWatch 541
将指标归因报告发布到 Amazon S3 541
通过 A/B 测试衡量推荐的影响 546
A/B 测试最佳实践 546
对 OpenSearch 搜索结果进行个性化设置 548
使用案例示例 549
Amazon Personalize 搜索排名插件的工作原理 549
其他信息 550
插件要求 550
对来自亚马逊 OpenSearch 服务的结果进行个性化设置 551
设置权限 552
安装插件 557
创建管道 558
应用插件 559
比较结果 560
监控插件 563
对开源 OpenSearch 的结果进行个性化设置 563
设置权限 565
在现有的 OpenSearch 集群上手动安装插件 567
使用脚本创建集群和安装插件 567

创建管道 568
应用插件 569
比较结果 570
监控插件 571
个性化搜索排名字段 571
管道指标示例 572
标注资源 575
指南和要求 576
附加信息 576
向 Amazon Personalize 资源添加标签 577
添加标签（控制台） 577
添加标签 (AWS CLI) 578
添加标签 (AWS SDKs) 578
从 Amazon Personalize 资源中删除标签 582
删除标签（控制台） 582
删除标签（AWS CLI） 582
删除标签（AWS SDKs） 583
在 IAM 策略中使用标签 583
常见问题 585
数据导入和管理 585
创建自定义解决方案和解决方案版本 586
模型部署（自定义市场活动） 586
建议 587
筛选建议 588
常见错误消息 589
数据导入和管理 589
创建解决方案和解决方案版本（自定义资源） 590
模型部署（自定义市场活动） 591
推荐器（域数据集组） 591
建议 591
筛选建议 591
使用指定资源 AWS CloudFormation 593
Amazon Persalize 和 CloudFormation 模板 593
Amazon Personalize 资源的示例 CloudFormation 模板 593
CreateDatasetGroup 594
CreateDataset 594

CreateSchema 595
CreateSolution 596
了解更多关于 CloudFormation 597
代码示例 598
Amazon Personalize 599
基本功能 600
Amazon Personalize Events 650
基本功能 651
Amazon Personalize Runtime 658
基本功能 659
安全性 668
数据保护 668
Amazon Personalize 中的数据加密 669
身份和访问管理 670
受众 671
使用身份进行身份验证 671
使用策略管理访问 672
如何配合使用 Amazon Personalize 与 IAM 673
Cross-service 混乱的副手预防 678
Identity-based 策略示例 679
问题排查 684
使用监控 CloudWatch 686
使用 Amazon Personalize 的 CloudWatch 指标 686
访问 Amazon Personalize 指标 687
创建警报 688
监控应用程序 689
CloudWatch 亚马逊 Personalize 的指标 690
使用记录亚马逊个性化 API 调用 AWS CloudTrail 694
Amazon Personalize 在 CloudTrail 694
示例：Amazon Personalize 日志文件条目 695
合规性验证 696
恢复能力 697
基础结构安全性 697
VPC 端点（AWS PrivateLink） 697
为 Amazon Personalize 创建接口 VPC 端点 698
为 Amazon Personalize 创建 VPC 端点策略 698

端点和限额 700
Amazon Personalize 端点和区域 700
合规 700
服务配额 700
请求提高配额 707
API 参考 709
操作 709
Amazon Personalize 712
Amazon Personalize Events 958
Amazon Personalize Runtime 973
数据类型 989
Amazon Personalize 992
Amazon Personalize Events 1148
Amazon Personalize Runtime 1160
常见错误类型 1165
常见参数 1167
文档历史记录 1170
..... mclxxxiii

什么是 Amazon Personalize？

Amazon Personalize 是一项完全托管的机器学习服务，可使用您的数据为用户生成物品建议。它还可以根据用户对某些物品或物品元数据的亲和度生成用户细分。

常见使用案例包括：

- 个性化视频流式传输应用程序 - 您可以使用预先配置或可自定义的 Amazon Personalize 资源，向您的流应用程序添加多种类型的个性化视频建议。例如，热门精选、更喜欢 X 和最受欢迎 视频建议。
- 向电子商务应用程序添加产品建议 - 您可以使用预配置或可自定义的 Amazon Personalize 资源，向您的零售应用程序添加多种类型的个性化产品建议。例如，为您推荐、经常一起购买 和浏览 X 的客户也浏览过 产品建议。
- 向您的应用程序添加实时下一步最佳操作建议 – 您可以使用可自定义的 Amazon Personalize 资源，根据用户的行为建议他们最有可能采取的操作。例如，您可以添加有关注册忠诚度计划、下载移动应用或注册促销电子邮件的实时建议。
- 创建个性化电子邮件 - 您可以使用可自定义的 Amazon Personalize 资源，为电子邮件列表中的所有用户生成批量建议。然后，您可以使用 AWS 服务或第三方服务向用户发送个性化电子邮件，以推荐您目录中的物品。
- 创建有针对性的营销活动 - 您可以使用 Amazon Personalize，生成最有可能与您目录中的物品进行交互的用户细分。然后，您可以使用 AWS 服务或第三方服务来创建有针对性的营销活动，以向不同的用户细分推广不同的物品。
- 个性化搜索结果 - 您可以使用可自定义的 Amazon Personalize 资源，为您的用户提供个性化搜索结果。例如，Amazon Personalize 可以对您生成的搜索结果进行OpenSearch重新排名。

对于大多数使用案例，Amazon Personalize 主要根据物品交互数据生成建议。物品交互数据来自用户与您目录中的物品进行的交互。例如，用户单击不同的物品。物品交互数据既可以来自 CSV 文件中的批量交互历史记录，也可以来自用户与您的目录交互时的实时事件。在某些情况下，Amazon Personalize 还会使用物品和用户中的数据，例如类型、价格或性别。对于下一个最佳操作场景，它使用操作和操作交互数据。

当你导入批量数据时，你可以使用 Amazon A SageMaker I Data Wrangler 从 40 多个来源导入数据，为亚马逊 Personalize 做好准备。有关更多信息，请参阅 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。

Amazon Personalize 包括用于实时个性化的 API 操作，以及用于批量建议和用户细分的批量操作。您可以快速开始使用针对业务域的使用案例优化的推荐器，也可以创建自己的可配置自定义资源。

主题

- Amazon Personalize 的定价
- 面向首次使用 Amazon Personalize 的用户的指南
- 相关 AWS 服务和解决方案
- Third-party 服务
- 了解详情
- 将 Amazon Personalize 与 AWS SDK

Amazon Personalize 的定价

使用 Amazon Personalize，无最低费用，无预先承诺。AWS 免费套餐为每个可用 AWS 区域提供每月最多 20 GB 的数据处理配额，每个符合条件 AWS 的地区提供最多 100 小时的培训时间，以及最多 180,000 份推荐请求。该免费套餐在使用的前两个月内有效。

有关费用和价格的完整列表，请参阅 Amazon Personalize 定价。

面向首次使用 Amazon Personalize 的用户的指南

如果您是首次使用 Amazon Personalize 的用户，以下资源可以为您提供相关帮助。

主题

- 借助 Magic Movie Machine 了解 Amazon Personalize
- 浏览本指南中的入门材料

借助 Magic Movie Machine 了解 Amazon Personalize

Magic Movie Machine 是一种交互式学习体验。它可以帮助您了解 Amazon Personalize 特征，并详细了解如何生成建议。有关简短的介绍，请观看下面的视频。然后试试 Magic Movie Machine。

开始使用 Amazon Personalize

浏览本指南中的入门材料

请查看以下部分，开始使用 Amazon Personalize。要获取列出了 Amazon Personalize 特征、要求和数据指南的清单，请参阅准备清单。

1. Amazon Personalize 的工作原理 - 本节介绍了 Amazon Personalize 工作流程，并引导您完成为用户创建个性化体验的步骤。本节还包括常见的 Amazon Personalize 术语及其定义。从本节开始，确保您在开始获取建议之前，充分理解 Amazon Personalize 工作流程和术语。
2. 设置 Amazon Personalize— 在本节中，您将设置您的 AWS 账户，设置使用亚马逊个性化所需的权限，并设置和软件开发工具包以使用和管理 Amazon Per AWS sonalize。 AWS CLI
3. 入门教程 - 在本节中，您将开始通过简单的电影数据集使用 Amazon Personalize。完成这些教程，获得实际操作 Amazon Personalize 的体验。您可以选择开始使用域数据集组或自定义数据集组：
   - 要开始创建域数据集组，请完成入门先决条件，然后开始域数据集组入门中的教程。
   - 要开始创建自定义数据集组，请完成入门先决条件，然后开始域数据集组入门中的教程。

完成入门练习后，您可以开始完成 Amazon Personalize 工作流程。有关步骤和文档链接，请参阅Amazon Personalize 工作流程。

相关 AWS 服务和解决方案

Amazon Personalize 与其他 AWS 服务和解决方案无缝集成。例如，您可以：

- 使用 Amazon SageMaker AI Data Wrangler（Data Wrangler）将来自 40 多个来源的数据导入到 Amazon Personalize 数据集中。Data Wrangler 是 Amazon A SageMaker I Studio 的一项功能，它为导入、准备、转换和分析数据提供了端到端解决方案。有关更多信息，请参阅 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。
- AWS Amplify 用于记录项目交互事件。Amplify 包括一个用于记录来自网络客户端应用程序的事件的 JavaScript库。而且，还包括一个用于记录服务器代码中的事件的库。有关更多信息，请参阅 Amplify 文档。
- 借助通过机器学习保持个性化体验，自动执行和安排 Amazon Personalize 任务。本 AWS 解决方案实施可自动执行 Amazon Personalize 工作流程，包括数据导入、解决方案版本训练和批量工作流程。

Third-party 服务

Amazon Personalize 可以很好地与各种第三方服务配合使用。

- Amplitude - 您可以使用 Amplitude 来跟踪用户操作，以帮助您了解用户的行为。有关使用 Amplitude 和 Amazon Personalize 的信息，请参阅以下 AWS 合作伙伴网络 (APN) 博客文章：使用 Amplitude 和 Amazon Personalize 衡量个性化的有效性。

- Braze - 您可以使用 Braze 向用户发送个性化电子邮件，以推荐您目录中的物品。Braze 是市场领先的消息收发平台（电子邮件、推送、SMS）。有关展示如何集成 Amazon Personalize 和 Braze 的研讨会，请参阅 Amazon Personalize 研讨会。
- Optimizely — 你可以使用 Optimizely 根据 A/B 亚马逊个性化推荐进行测试。
- Segment - 您可以使用 Segment 将您的数据发送到 Amazon Personalize。有关将 Segment 与 Amazon Personalize 集成的更多信息，请参阅 Amazon Personalize 目标。

有关合作伙伴的完整列表，请参阅 Amazon Personalize 合作伙伴。

了解详情

以下资源提供了有关 Amazon Personalize 的更多信息：

- 要获取快速参考以帮助您确定 Amazon Personalize 是否适合您的使用案例，请参阅 Amazon Personalize 示例存储库中的 Amazon Personalize 备忘单。
- 有关如何使用 Amazon Personalize 的一系列视频，请参阅上的 A mazon Personalize Deep Dive 视频系列 YouTube。
- 有关深入的教程和代码示例，请参阅亚马逊个性化 GitHub 示例存储库。

将 Amazon Personalize 与 AWS SDK

AWS 软件开发套件 (SDK) 可用于许多流行的编程语言。每个软件开发工具包都提供 API、代码示例和文档，使开发人员能够更轻松地以其首选语言构建应用程序。

SDK 文档	代码示例
适用于 C++ 的 AWS SDK	适用于 C++ 的 AWS SDK 代码示例
AWS CLI	AWS CLI 代码示例
适用于 Go 的 AWS SDK	适用于 Go 的 AWS SDK 代码示例
适用于 Java 的 AWS SDK	适用于 Java 的 AWS SDK 代码示例
适用于 JavaScript 的 AWS SDK	适用于 JavaScript 的 AWS SDK 代码示例
适用于 Kotlin 的 AWS SDK	适用于 Kotlin 的 AWS SDK 代码示例

SDK 文档	代码示例
适用于 .NET 的 AWS SDK	适用于 .NET 的 AWS SDK 代码示例
适用于 PHP 的 AWS SDK	适用于 PHP 的 AWS SDK 代码示例
AWS Tools for PowerShell	AWS Tools for PowerShell 代码示例
AWS 适用于 Python 的 SDK v4	AWS 适用于 Python 的 SDK v4 代码示例
适用于 Python (Boto3) 的 AWS SDK	适用于 Python (Boto3) 的 AWS SDK 代码示例
适用于 Ruby 的 AWS SDK	适用于 Ruby 的 AWS SDK 代码示例
适用于 Rust 的 AWS SDK	适用于 Rust 的 AWS SDK 代码示例
适用于 SAP ABAP 的 AWS SDK	适用于 SAP ABAP 的 AWS SDK 代码示例
适用于 Swift 的 AWS SDK	适用于 Swift 的 AWS SDK 代码示例

示例可用性

找不到所需的内容？ 通过使用此页面底部的提供反馈链接请求代码示例。

Amazon Personalize 和生成式人工智能

Amazon Personalize 与生成式人工智能配合得很好。Amazon Personalize 内容生成器在生成式人工智能的帮助下，可以在相关物品的批量建议中添加吸引人的主题。内容生成器是一项由 Amazon Personalize 管理的生成式人工智能功能。

还可以使用 Amazon Personalize 建议将 Amazon Personalize 与您的生成式人工智能工作流程集成并增强用户体验。例如，您可以在生成式人工智能提示中添加建议，以创建针对每个用户的兴趣量身定制的营销内容。您还可以为建议的内容生成简明摘要，或者通过聊天机器人建议产品或内容。

以下视频演示了如何使用 Amazon Personalize 和生成式人工智能来增强推荐。

使用 Amazon Personalize 和生成式人工智能增强推荐

以下 Amazon Personalize 特征使用生成式人工智能，或者可以帮助您构建生成式人工智能解决方案来创建个性化内容。有关演示如何将 Amazon Personalize 与生成式人工智能结合使用的示例 Jupyter Notebook，请参阅 Amazon Personalize 示例存储库中的生成式人工智能和 Amazon Personalize。

主题

- 内容生成器中带有主题的建议
- 建议元数据
- 预配置的用于个性化的 LangChain 代码

内容生成器中带有主题的建议

Amazon Personalize 内容生成器可以向批量建议添加描述性主题。内容生成器是一项由 Amazon Personalize 管理的生成式人工智能功能。

当您获得带有主题的批量建议时，Amazon Personalize 内容生成器会为每组相似的物品添加一个描述性主题。例如，如果您收到早餐食品的相似物品建议，Amazon Personalize 可能会生成一个主题，例如快起床吧或早晨必备品。您可以使用该主题来替换通用的轮播标题，例如经常一起购买。或者，您可以将该主题纳入促销电子邮件或营销活动中，以获得新的菜单选项。

要生成主题，您需要将数据导入物品交互数据集和物品数据集，使用 Similar-Items 配方创建自定义解决方案，然后生成批量建议。您的物品数据必须包括物品描述和标题信息。详细的物品描述和标题可帮助内容生成器创建更准确、更具吸引力的主题。

- 有关 Amazon Personalize 工作流程的信息，请参阅Amazon Personalize 工作流程详细信息。

- 有关批量推荐的信息，请参阅获取批量物品推荐和获取批量用户细分。
- 有关生成带有主题的物品推荐的信息，请参阅内容生成器中带有主题的批量建议。

建议元数据

获得建议后，您可以让 Amazon Personalize 从您的物品数据集中返回有关每个建议物品的元数据。您可以将此元数据以及 Amazon Personalize 建议添加到生成式人工智能提示中，以生成更具吸引力的内容。

例如，您可以使用生成式人工智能来创建营销电子邮件。您可以使用 Amazon Personalize 建议及其元数据（例如电影类型）作为生成式人工智能的提示工程的一部分。借助个性化提示，您可以使用生成式人工智能来创建针对每个客户的兴趣量身定制的吸引人的营销电子邮件。

要获取建议元数据，您需要先完成 Amazon Personalize 工作流程以导入数据并创建域或自定义资源。在创建 Amazon Personalize 推荐器或市场活动时，启用在建议中包含元数据的选项。获得建议后，您可以指定要包含哪些列的物品数据。

- 有关 Amazon Personalize 工作流程的信息，请参阅Amazon Personalize 工作流程详细信息。
- 有关为推荐器启用元数据的信息，请参阅在建议中启用元数据（域资源）。
- 有关为市场活动启用元数据的信息，请参阅在建议中启用元数据（自定义资源）。
- 有关如何将 Amazon Personalize 与生成式人工智能结合使用来创建营销活动的更多信息，请参阅使用 Amazon Personalize 和生成式人工智能提升您的营销解决方案。

预配置的用于个性化的 LangChain 代码

LangChain 是一个框架，用于开发由语言模型提供支持的应用程序。它包含专为 Amazon Personalize 构建的代码。您可以使用此代码将 Amazon Personalize 建议与生成式人工智能解决方案集成。

例如，您可以使用以下代码向您的链中添加针对用户的 Amazon Personalize 建议。

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

recommender_arn="RECOMMENDER ARN"

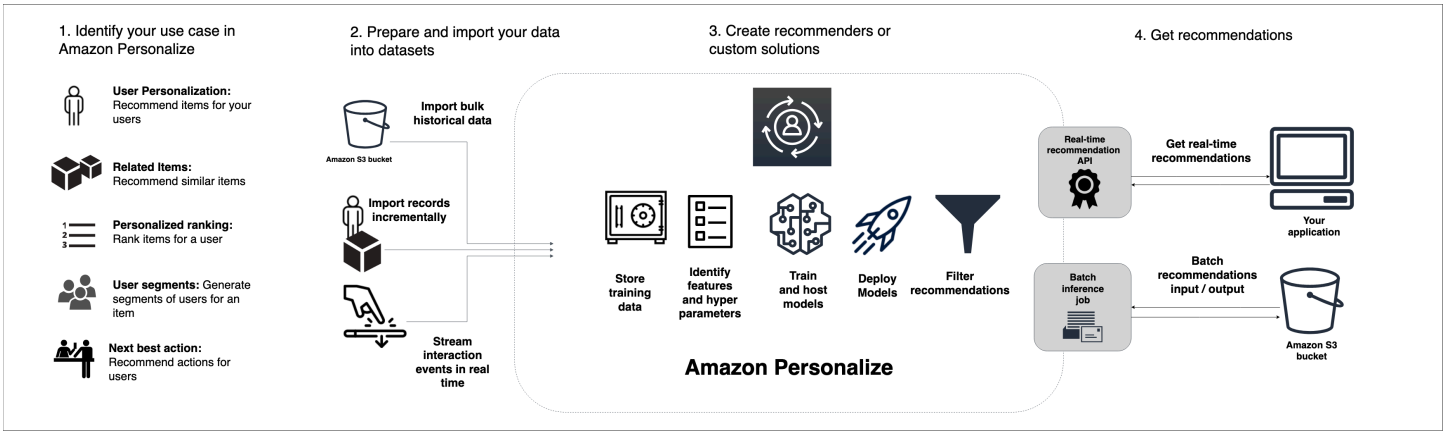
bedrock_llm = Bedrock(model_id="anthropic.claude-v2", region_name="us-west-2")
```

```
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)
```

- 有关开始使用 LangChain 的信息，请参阅 LangChain 文档中简介。
- 有关使用专为 Amazon Personalize 构建的 LangChain 代码的信息，包括更高级的代码示例，请参阅 AWS 示例存储库中的 Amazon Personalize LangChain 扩展。

Amazon Personalize 的工作原理

Amazon Personalize 使用您的数据来训练基于域或可自定义的建议模型。您可以在应用程序中使用私有建议 API 来请求实时建议。Amazon Personalize 还支持用于获取物品建议和用户细分的批量工作流程。



主题

- Amazon Personalize 工作流程详细信息
- Amazon Personalize 术语
- Amazon Personalize 可以使用的数据类型

Amazon Personalize 工作流程详细信息

Amazon Personalize 工作流程如下。要获取列出了 Amazon Personalize 特征、要求和数据指南的清单，请参阅准备清单。

1. 将您的应用场景与 Amazon Personalize 资源进行匹配 - Amazon Personalize 提供基于域的资源和为不同应用场景配置的自定义资源。当您将自己的应用场景与 Amazon Personalize 资源进行匹配时，请注意其数据要求。在您选择应用场景或配方后，此信息有助于准备数据。
2. 准备训练数据 - 根据域应用场景或自定义配方的数据要求，在 CSV 文件中准备批量训练数据。Amazon Personalize 可以使用物品交互、物品、用户、操作和操作交互数据，具体取决于您的应用场景或配方。如果您没有批量数据，则可以使用单独的导入操作来收集数据和直播事件，直到满足 Amazon Personalize 训练要求以及域应用场景或配方的数据要求。
3. 为您的数据创建架构 JSON 文件 - 为您要导入的每种数据类型创建架构 JSON 文件。这些文件概述了数据的结构和内容，包括列名及其数据类型。

4. 创建数据集组 - 数据集组是存放 Amazon Personalize 资源的容器。您可为 VIDEO_ON_DEMAND 或 ECOMMERCE 域创建具有预配置资源的域数据集组。或者您可以创建自定义数据集组，并且只能创建自定义资源。
5. 创建架构和数据集 - 架构会将您的数据结构告知给 Amazon Personalize，并允许 Amazon Personalize 解析数据。数据集是 Amazon Personalize 中用于训练数据的容器。
6. 将训练数据导入到数据集 - 导入准备好的交互、物品、用户、操作或操作交互记录。您可以批量导入记录，也可以单独导入记录。
7. 训练和部署模型 - 要在 VIDEO_ON_DEMAND 或 ECOMMERCE 域训练和部署模型，可以创建域推荐器。对于自定义资源，创建自定义解决方案和解决方案版本。对于实时建议，您可以在市场活动中部署解决方案版本。
   - 有关创建域推荐器的信息，请参阅域推荐器。
   - 有关创建和部署自定义资源的信息，请参阅自定义资源。
8. 获取建议 - 使用您的推荐器或自定义市场活动来获取建议。您可以使用筛选器，在建议中包含或排除某些类型的物品。有关更多信息，请参阅 筛选建议和用户细分。使用自定义资源时，无需创建市场活动也可以获取批量推荐或用户细分。
9. 记录实时事件 - 记录客户与推荐交互时的实时事件。这会生成您的交互数据，并使您的数据保持最新状态。并且它会告知 Amazon Personalize 您的用户当前的兴趣，从而提高建议的相关性。

首次完成 Amazon Personalize 工作流程后，确保数据保持最新状态，并定期重新训练使用手动训练的任何自定义解决方案。这样，您的模型能够从用户的最新活动中学习，并维持和提高建议的相关性。有关更多信息，请参阅 维护建议相关性。

Amazon Personalize 术语

本节介绍了 Amazon Personalize 中使用的术语。

主题

- 数据导入和管理
- 训练
- 模型部署和建议

数据导入和管理

以下术语与在 Amazon Personalize 中导入、导出数据及其格式设置有关。

操作数据集

用于存放操作元数据的容器。操作是指您可能想向用户推荐的交互或创收活动，例如安装移动应用程序，或加入忠诚度计划。操作的元数据可能包括操作的到期时间戳、值、重复频率数据和分类元数据。只有 Next-Best-Action 配方 才使用此数据类型。

操作交互数据集

用于保存从用户和操作之间的交互中收集的历史数据和实时数据的容器。每个操作交互都包含 userID、actionID、时间戳、事件类型以及有关交互的任何其它数据，如分类元数据。只有 Next-Best-Action 配方 才使用此数据类型。

上下文元数据

您在事件（例如单击）发生时收集的有关用户浏览上下文（例如使用的设备或位置）的交互数据。上下文元数据可以改善新用户和现有用户的建议相关性。

数据集

用于存放您上传到 Amazon Personalize 的数据的容器。Amazon Personalize 数据集有五种类型：用户、项目、项目交互数据集、操作交互数据集和操作。

数据集组

Amazon Personalize 资源的容器，包括数据集、域推荐器和自定义资源。数据集组将资源组织到独立的集合中，在其中，一个数据集组中的资源不会影响任何其他数据集组中的资源。数据集组可以是域数据集组，也可以是自定义数据集组。

域数据集组

一个数据集组，包含针对不同业务域和使用案例的预配置资源。Amazon Personalize 负责管理训练模型和部署的生命周期。创建域数据集组时，您可以选择业务域，导入数据，并为每个使用案例创建推荐器。您可以在应用程序中使用推荐器，通过 GetRecommendations 操作获取建议。

如果您从域数据集组开始，则仍然可以添加自定义资源，例如解决方案和解决方案版本，这些版本使用自定义使用案例食谱进行训练。

自定义数据集组

仅包含自定义资源的数据集组，包括解决方案、解决方案版本、筛选器、市场活动和批量推理作业。您可以使用市场活动，通过 GetRecommendations 操作获取建议。您负责管理训练模型和部署的生命周期。如果您从自定义数据集组开始，则以后无法将其与域相关联。相反，创建新的域数据集组。

数据集导出作业

一种记录导出工具，可将数据集中的记录输出到 Amazon S3 存储桶中的一个或多个 CSV 文件。输出 CSV 文件包括标题行，其列名称与数据集架构中的字段相匹配。

数据集导入作业

一种批量导入工具，可使用来自 Amazon S3 存储桶中 CSV 文件的数据填充您的 Amazon Personalize 数据集。

事件

您记录并上传到 Amazon Personalize 物品交互数据集的用户操作（如单击、购买或视频观看）。您可以通过 Amazon Personalize 控制台以增量方式从 CSV 文件中实时批量导入事件。

显式展示

您手动添加到 Amazon Personalize 物品交互数据集的物品列表。不同于隐式展示（Amazon Personalize 自动从建议数据中派生此类展示），您可以选择要在显式展示中包含的物品。

隐式展示

您的应用程序向用户显示的建议。不同于显式展示（手动将此类展示添加到物品交互数据集），Amazon Personalize 自动从您的建议数据派生隐式展示。

展示数据

当用户与特定物品进行交互（例如单击、观看或购买等）时，您向其展示的物品列表。Amazon Personalize 根据用户选择或忽略同一物品的频率，使用展示数据计算新物品对用户的相关性。

交互数据集

用于保存从用户和物品（称为事件）之间的交互中收集的历史数据和实时数据的容器。交互数据包括事件类型数据和上下文元数据。

物品数据集

用于保存物品相关元数据（例如价格、类型或可用性）的容器。

重复频率

一种可以导入到操作数据集的操作元数据。重复频率数据根据用户在您的操作交互数据集中的历史记录，指定 Amazon Personalize 在用户与特定操作交互后应等待多少天才能建议该操作。

架构

Apache Avro 格式的 JSON 对象，用于向 Amazon Personalize 告知您的数据结构。Amazon Personalize 使用您的架构来解析您的数据。

用户数据集

用于保存用户相关元数据（例如年龄、性别或会员资格）的容器。

训练

以下术语与在 Amazon Personalize 中训练模型有关。

物品间相似度 (SIMS) 配方

RELATED_ITEMS 食谱，使用交互数据集中的数据为类似于指定物品的物品生成建议。SIMS 食谱根据用户与物品进行交互的方式而不是匹配物品元数据（如价格或颜色）计算相似度。

item-affinity

USER_SEGMENTATION 配方，它使用来自物品交互数据集和物品数据集的数据，根据用户与物品交互的可能性为您指定的每个物品创建用户细分。

item-attribute-affinity

USER_SEGMENTATION 配方，它使用来自物品交互数据集和物品数据集的数据，根据用户与带有属性的物品交互的可能性为您指定的每个物品属性创建用户细分。

Next-Best-Action 配方

此配方为您的用户生成下一个最佳操作的实时推荐。用户的下一个最佳操作是他们最有可能采取的操作。例如，注册您的忠诚度计划、下载您的应用或申请信用卡。有关更多信息，请参阅 Next-Best-Action 配方。

Personalized-Ranking-v2 配方

PERSONALIZED_RANKING 食谱，根据特定用户的预测兴趣级别对您提供的一系列物品进行排名。此配方使用基于转换器的架构来训练模型，该模型可以从物品交互数据、物品元数据和用户元数据中学习。使用 Personalized-Ranking-v2 配方，可对针对特定用户个性化的物品或搜索结果的精选列表顺序进行个性化设置。它可对多达 500 万个物品进行训练并生成更相关的推荐，延迟比之前版本更低。

个性化-排名配方

PERSONALIZED_RANKING 食谱，根据特定用户的预测兴趣级别对您提供的一系列物品进行排名。使用 personalized-ranking 食谱，可对针对特定用户个性化的物品或搜索结果的精选列表顺序进行个性化设置。

热门程度-计数配方

USER_PERSONALIZATION 配方，用于推荐与唯一用户交互最多的物品。

推荐器

用于生成建议的域数据集组工具。您可以为域数据集组创建推荐器，并在应用程序中使用它通过 GetRecommendations API 获取实时建议。创建推荐器时，您需指定一个使用案例，然后 Amazon Personalize 通过使用案例的最佳配置，训练支持推荐器的模型。

recipe

一种预先配置的 Amazon Personalize 算法，用于预测用户将与之进行交互的物品（对于 USER_PERSONALIZATION 食谱），或计算与用户表示感兴趣的特定物品类似的物品（对于 RELATED_ITEMS 食谱），或根据特定用户的预测兴趣对您提供的一系列物品进行排名（对于 PERSONALIZED_RANKING 食谱）。

解决方案

Amazon Personalize 用来生成建议的食谱、自定义参数和经过训练的模型（解决方案版本）。

解决方案版本

作为 Amazon Personalize 解决方案的一部分创建的经过训练的模型。您可以在市场活动中部署解决方案版本以激活用于请求建议的个性化 API。

训练模式

创建解决方案版本时要执行的训练的范围。有以下两种不同的模式：“完整”和“更新”。完整模式基于数据集组中数据集的全部训练数据，创建一个全新的解决方案版本。更新模式以增量方式更新现有解决方案版本，以推荐自上次训练以来添加的新物品。

> **Note**
>
> 使用 User-Personalization-v2、User-Personalization 或 Next-Best-Action 时，Amazon Personalize 自动更新在完整训练模式下训练的最新解决方案版本。请参阅自动更新。

User-Personalization-v2 配方

USER_PERSONALIZATION 配方根据用户的偏好推荐用户将与之交互的物品。此配方使用基于转换器的架构来训练模型，该模型可以从物品交互数据、物品元数据和用户元数据中学习。它可对多达 500 万个物品进行训练并生成更相关的推荐，延迟比之前版本更低。

User-Personalization 食谱

基于分层递归神经网络 (HRNN) 的 USER_PERSONALIZATION 食谱，可预测用户将与之交互的物品。用户-个性化配方可以使用物品浏览和展示数据生成有关新物品的推荐。

模型部署和建议

以下术语与部署和使用模型生成推荐相关。

操作优化期

Amazon Personalize 在预测用户最有可能采取的操作时使用的时间段。例如，如果操作优化期为 14 天，Amazon Personalize 预测用户在接下来的 14 天内最有可能采取的操作。使用 Next-Best-Action 配方 创建解决方案时，您可以配置操作优化期。

批量推理作业

一种工具，可从 Amazon S3 存储桶导入您的批量输入数据，使用您的解决方案版本生成建议，并将建议导出到 Amazon S3 存储桶。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。使用批量推理作业，针对不需要实时更新的大型数据集获取建议。

批量细分作业

一种工具，可从 Amazon S3 存储桶导出批量输入数据，使用您的解决方案版本创建用户细分，并将用户细分导出到 Amazon S3 存储桶。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。通过由 USER_SEGMENTATION 食谱支持的解决方案使用批量细分作业，根据用户与不同物品或具有不同物品属性的物品交互的可能性来创建用户细分。

市场活动

部署的解决方案版本（经过训练的模型），具有预调配的专用事务容量，用于为您的应用程序用户创建实时建议。创建市场活动后，您可以使用 `getRecommendations` 或 `getPersonalizedRanking` API 操作来获取建议。

物品浏览

在浏览中，建议包括一些通常不太可能向用户建议的物品或操作，例如新物品或操作、交互量很少的物品或操作，或者根据用户以前的行为与用户不太相关的物品或操作。

指标归因

您用来衡量物品推荐所产生影响的工具。指标归因可根据您导入的物品交互和物品数据以及您指定的指标创建报告。例如，用户观看的影片总时长或单击事件的总数。

建议

Amazon Personalize 预测用户将与之交互的物品列表。根据所使用的 Amazon Personalize 食谱，建议可以是物品列表（使用 USER_PERSONALIZATION 食谱和 RELATED_ITEMS 食谱），也可以是您提供的一系列物品的排名（使用 PERSONALIZED_RANKING 食谱）。

用户细分

Amazon Personalize 预测用户将与您的目录交互的用户列表。根据所使用的 USER_SEGMENTATION 食谱，您可以根据物品（Item-Affinity 食谱）和物品元数据（Item-Attribute-Affinity 食谱）创建用户细分。您可以使用批量细分作业来创建用户细分。

Amazon Personalize 可以使用的数据类型

以下主题介绍了您可以导入 Amazon Personalize 的不同类型的数据。

主题

- 交互数据
- 物品数据
- 用户数据
- 操作数据
- 操作交互数据

交互数据

交互 是您记录然后将其作为训练数据导入的事件。Amazon Personalize 主要根据交互数据生成建议。交互数据可包含以下内容：

- 事件类型和事件值数据
- 上下文元数据
- 展示数据

您可以将交互数据导入到物品交互数据集中。有关物品交互数据集的更多详细信息，请参阅物品交互数据。

物品数据

Amazon Personalize 可以使用的物品元数据包括以下内容：

- 有关每个物品的数值数据，例如其价格。
- 有关每个物品的分类元数据，例如物品的类型或颜色。
- 每个物品的创建时间戳数据。
- 非结构化文本元数据，例如产品描述或电影提要。

您可以将有关物品的元数据导入到物品 数据集中。有关物品数据集的更多信息，请参阅物品元数据。

用户数据

Amazon Personalize 可以使用的用户元数据包括以下内容：

- 有关每个用户的数值数据，例如其年龄。
- 有关每个用户的分类元数据，例如其性别或会员资格状态。

您可以将有关用户的元数据导入到用户 数据集中。有关用户数据集的更多信息，请参阅用户元数据。

操作数据

Amazon Personalize 可以使用的操作数据包括以下内容：

- 每个操作的商业价值或重要性。
- 每个操作的分类元数据，例如季节性或操作排他性。
- 操作到期时间戳数据，用于指定 Amazon Personalize 何时应停止建议每个操作。
- 在用户与每个操作进行交互后，Amazon Personalize 应等待，然后再建议每个操作的重复频率数据。

您可以将有关您的操作的数据导入到操作数据集中。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。有关操作数据集的更多信息，请参阅操作元数据。

操作交互数据

Amazon Personalize 可以从用户与操作的交互中使用的数据包括以下内容：

- 事件类型数据
- 分类元数据

您可以将交互数据导入操作交互数据集。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。有关操作交互数据集的更多详细信息，请参阅操作交互数据。

设置 Amazon Personalize

在使用 Amazon Personalize 之前，您必须拥有具有管理用户的 Amazon Web Services (AWS) 账户。设置所需权限后，您可以通过 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或软件开发工具包访问 Amazon Per AWS sonalize。

主题

- 注册获取 AWS 账户
- 区域和端点
- 设置权限
- 设置 AWS CLI
- 设置 AWS 软件开发工具包

注册获取 AWS 账户

要开始使用 AWS，你需要一个 AWS 账户。有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》 AWS 账户中的入门指南。

区域和端点

终端节点是作为 Web 服务入口点的 URL。每个终端节点都与特定的 AWS 区域关联。请注意 Amazon Personalize 控制台 AWS CLI、和 Amazon Personalize SDK 的默认区域，因为给定活动的所有亚马逊个性化组件（数据集、解决方案、活动、事件跟踪器）都必须在同一区域创建。有关 Amazon Personalize 支持的区域和端点，请参阅区域和端点。

设置权限

您必须向用户、组或角色授予与 Amazon Personalize 资源进行交互的权限。而且，还必须向 Amazon Personalize 授予以下权限：访问您在 Amazon Personalize 中创建的资源并代表您执行任务。

设置 权限

1. 向 Amazon Personalize 授予以下权限：访问您在 Amazon Personalize 中的资源并代表您执行任务。请参阅向 Amazon Personalize 授予访问资源的权限。
2. 向您的用户、组或角色授予以下权限：与 Amazon Personalize 资源进行交互并将您的服务角色传递给 Amazon Personalize。请参阅向用户授予访问 Amazon Personalize 的权限。

3. 修改 Amazon Personalize 服务角色的信任策略，以防出现混淆代理人问题。有关信任关系策略示例，请参阅Cross-service 混乱的副手预防。有关修改角色信任策略的信息，请参阅修改角色。
4. 如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
5. 完成向 Amazon Personalize 授予访问 Amazon S3 资源的权限中的步骤，使用 IAM 和 Amazon S3 存储桶策略向 Amazon Personalize 授予访问您 Amazon S3 资源的权限。

主题

- 向 Amazon Personalize 授予访问资源的权限
- 向用户授予访问 Amazon Personalize 的权限
- 向 Amazon Personalize 授予访问 Amazon S3 资源的权限
- 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限

向 Amazon Personalize 授予访问资源的权限

要向 Amazon Personalize 授予访问您资源的权限，您需要创建一个 IAM 策略，以便为 Amazon Personalize 提供对您的 Amazon Personalize 资源的完全访问权限。或者您可以使用 AWS 托管`AmazonPersonalizeFullAccess`策略。 `AmazonPersonalizeFullAccess`提供的权限超出了必要的范围。我们建议创建一个仅授予必要权限的新 IAM 策略。有关托管策略的更多信息，请参阅AWS 托管策略。

创建策略之后，为 Amazon Personalize 创建一个 IAM 角色并向其附加新策略。

主题

- 为 Amazon Personalize 创建新 IAM 策略
- 为 Amazon Personalize 创建 IAM 角色

为 Amazon Personalize 创建新 IAM 策略

创建一个 IAM 策略，以便为 Amazon Personalize 提供对 Amazon Personalize 资源的完全访问权限。

使用 JSON 策略编辑器创建策略

1. 登录 AWS 管理控制台 并打开 IAM 控制台，网址为https://console.aws.amazon.com/iam/。

2. 在左侧的导航窗格中，选择策略。

   如果这是您首次选择策略，则会显示欢迎访问托管式策略页面。选择开始使用。

3. 在页面的顶部，选择创建策略。
4. 在策略编辑器部分，选择 JSON 选项。
5. 输入以下 JSON 策略文档：

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

6. 选择下一步。

> **Note**
>
> 您可以随时在可视化和 JSON 编辑器选项卡之间切换。不过，如果您进行更改或在可视化编辑器中选择下一步，IAM 可能会调整策略结构以针对可视化编辑器进行优化。有关更多信息，请参阅《IAM 用户指南》中的调整策略结构。

7. 在查看并创建页面上，为您要创建的策略输入策略名称和描述（可选）。查看此策略中定义的权限以查看策略授予的权限。
8. 选择创建策略可保存新策略。

为 Amazon Personalize 创建 IAM 角色

要使用 Amazon Personalize，您必须为 Amazon Personalize 创建 AWS Identity and Access Management 服务角色。服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。为 Amazon Personalize 创建服务角色后，根据需要向该角色授予其他服务角色权限中列出的其他权限。

为 Personalize 创建服务角色（IAM 控制台）

1. 登录 AWS 管理控制台 并打开 IAM 控制台，网址为https://console.aws.amazon.com/iam/。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择创建角色。
3. 对于 Trusted entity type（可信实体类型），选择 AWS 服务。
4. 对于服务或使用案例，选择 Personalize，然后选择个性化应用场景。
5. 选择下一步。
6. 选择您在前一个过程中创建的策略。
7. （可选）设置权限边界。这是一项高级功能，可用于服务角色，但不可用于服务相关角色。
   a. 打开设置权限边界部分，然后选择使用权限边界控制最大角色权限。

      IAM 包含您账户中的 AWS 托管策略和客户托管策略列表。
   b. 选择要用于权限边界的策略。
8. 选择下一步。
9. 输入有助于识别角色的作用的角色名称或者角色名称后缀。

   > **Important**
   >
   > 命名角色时，请注意以下事项：
   >
   > - 角色名称在您内部必须是唯一的 AWS 账户，并且不能因大小写而变得唯一。
   >
   >   例如，不要同时创建名为 **PRODROLE** 和 **prodrole** 的角色。当角色名称在策略中使用或者作为 ARN 的一部分时，角色名称区分大小写，但是当角色名称在控制台中向客户显示时（例如，在登录期间），角色名称不区分大小写。
   > - 创建角色后，您无法编辑该角色的名称，因为其他实体可能会引用该角色。
10. （可选）对于描述，输入角色的描述。
11. （可选）要编辑角色的使用案例和权限，请在步骤 1：选择可信实体或步骤 2：添加权限部分中选择编辑。
12. （可选）为了帮助识别、组织或搜索角色，请以键值对形式添加标签。有关在 IAM 中使用标签的更多信息，请参阅 IAM 用户指南中的AWS Identity and Access Management 资源标签。
13. 检查该角色，然后选择创建角色。

为 Amazon Personalize 创建角色后，您就可以向其授予访问您的 Amazon S3 存储桶和任何 AWS KMS 密钥的权限了。

其他服务角色权限

在创建角色并向该角色授予访问 Amazon Personalize 中资源的权限后，执行以下操作：

1. 修改 Amazon Personalize 服务角色的信任策略，以防出现混淆代理人问题。有关信任关系策略示例，请参阅Cross-service 混乱的副手预防。有关修改角色信任策略的信息，请参阅修改角色。
2. 如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

向用户授予访问 Amazon Personalize 的权限

要向用户授予访问 Amazon Personalize 的权限，您需要创建一个 IAM 策略，以便授予访问 Amazon Personalize 资源的权限并将您的服务角色传递给 Amazon Personalize。然后，向您的用户、组或角色添加权限时使用该策略。

为您的用户创建新 IAM 策略

创建一个 IAM 策略，该策略将向 Amazon Personalize 授予对您的 Amazon Personalize 资源的完全访问权限，并授予将您的服务角色（在为 Amazon Personalize 创建 IAM 角色中创建）传递给 Amazon Personalize 的 `PassRole` 权限。

使用 JSON 策略编辑器创建策略

1. 登录 AWS 管理控制台 并打开 IAM 控制台，网址为https://console.aws.amazon.com/iam/。
2. 在左侧的导航窗格中，选择策略。

   如果这是您首次选择策略，则会显示欢迎访问托管式策略页面。选择开始使用。
3. 在页面的顶部，选择创建策略。
4. 在策略编辑器部分，选择 JSON 选项。
5. 输入以下 JSON 策略文档：

```
{
    "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. 选择下一步。

> **Note**
>
> 您可以随时在可视化和 JSON 编辑器选项卡之间切换。不过，如果您进行更改或在可视化编辑器中选择下一步，IAM 可能会调整策略结构以针对可视化编辑器进行优化。有关更多信息，请参阅《IAM 用户指南》中的调整策略结构。

7. 在查看并创建页面上，为您要创建的策略输入策略名称和描述（可选）。查看此策略中定义的权限以查看策略授予的权限。
8. 选择创建策略可保存新策略。

要仅授予在 Amazon Personalize 中执行任务所需的权限，请修改之前的策略，使其仅包含您的用户所需的操作。有关 Amazon Personalize 操作的完整列表，请参阅 Amazon Personalize 的操作、资源和条件键。

提供对 Amazon Personalize 的访问权限

当您向用户提供权限时，附加新的 IAM 策略。

要提供访问权限，请为您的用户、组或角色添加权限：

- 用户和群组位于 AWS IAM Identity Center：

  创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。
- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

向 Amazon Personalize 授予访问 Amazon S3 资源的权限

要向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限，请执行以下操作：

1. 如果您还没有设置，请按照设置权限中的步骤设置权限，以便 Amazon Personalize 代表您访问您在 Amazon Personalize 中的资源。
2. 向 Amazon Personalize 服务角色（参见为 Amazon Personalize 创建 IAM 角色）附加允许访问 Amazon S3 存储桶的策略。有关更多信息，请参阅 将 Amazon S3 策略附加到您的 Amazon Personalize 服务角色中。
3. 将存储桶策略附加到包含数据文件的 Amazon S3 存储桶，以便 Amazon Personalize 访问这些文件。有关更多信息，请参阅 将 Amazon Personalize 访问策略附加到您的 Amazon S3 存储桶中。
4. 如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

> **Note**
>
> 由于 Amazon Personalize 不与通信 AWS VPCs，因此 Amazon Personalize 无法与仅允许 VPC 访问的 Amazon S3 存储桶进行交互。

主题

- 将 Amazon S3 策略附加到您的 Amazon Personalize 服务角色中
- 将 Amazon Personalize 访问策略附加到您的 Amazon S3 存储桶中

将 Amazon S3 策略附加到您的 Amazon Personalize 服务角色中

要将 Amazon S3 策略附加到您的 Amazon Personalize 角色中，请执行以下操作：

1. 登录 IAM 控制台 (https://console.aws.amazon.com/iam/)。
2. 在导航窗格中选择 Policies (策略)，然后选择 Create policy (创建策略)。
3. 选择“JSON”选项卡，然后更新策略，如下所示。将 `amzn-s3-demo-bucket` 替换为您的存储桶的名称。您可以对数据集导入作业或数据删除作业使用以下策略。如果您使用的是批量工作流程或正在创建数据集导出作业，则 Amazon Personalize 需要额外权限。请参阅批量工作流程的服务角色策略或用于导出数据集的 Amazon S3 存储桶策略。

   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. 选择下一步：标签。（可选）添加任何标签并选择查看。
5. 为策略命名。
6. （可选）对于描述，输入描述此策略的简短句子，例如，**Allow Amazon Personalize to access its Amazon S3 bucket.**

7. 选择创建策略。
8. 在导航窗格中，选择角色，然后选择您为 Amazon Personalize 创建的角色。请参阅为 Amazon Personalize 创建 IAM 角色。
9. 对于权限，选择附加策略。
10. 要显示列表中的策略，请在 Filter policies (筛选策略) 筛选框中键入部分策略名称。
11. 选中在该过程前面创建的策略旁边的复选框。
12. 选择附加策略。

在您的角色准备好用于 Amazon Personalize 之前，您还必须将存储桶策略附加到包含您的数据的 Amazon S3 存储桶。请参阅将 Amazon Personalize 访问策略附加到您的 Amazon S3 存储桶中。

批量工作流程的服务角色策略

要完成批量工作流程，Amazon Personalize 需要权限来访问文件并将其添加到 Amazon S3 存储桶。按照上述步骤将以下策略附加到您的 Amazon Personalize 角色中。将 `amzn-s3-demo-bucket` 替换为您的存储桶的名称。有关批量工作流程的更多信息，请参阅获取批量物品推荐或获取批量用户细分。

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/*"
            ]
        }
    ]
}
```

用于导出数据集的服务角色策略

要导出数据集，您的 Amazon Personalize 服务角色需要权限来针对 Amazon S3 存储桶使用 `PutObject` 和 `ListBucket` 操作。以下示例策略向 Amazon Personalize 授予 `PutObject` 和 `ListBucket` 权限。将 `amzn-s3-demo-bucket` 替换为存储桶的名称，并将该策略附加到您的 Amazon Personalize 服务角色中。有关将策略附加到服务角色的信息，请参阅将 Amazon S3 策略附加到您的 Amazon Personalize 服务角色中。

```
{
    "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/*"
            ]
        }
    ]
}
```

将 Amazon Personalize 访问策略附加到您的 Amazon S3 存储桶中

Amazon Personalize 需要权限才能访问 S3 存储桶。您可以对数据集导入作业或数据删除作业使用以下策略。将 `amzn-s3-demo-bucket` 替换为您的存储桶的名称。有关批量工作流程，请参阅针对批量工作流程的 Amazon S3 存储桶策略。

有关 Amazon S3 存储桶策略的更多信息，请参阅如何添加 S3 存储桶策略？。

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/*"
            ]
        }
    ]
}
```

针对批量工作流程的 Amazon S3 存储桶策略

对于批量工作流程，Amazon Personalize 需要权限来访问文件并将其添加到 Amazon S3 存储桶。将以下策略附加到您的存储桶：将 `amzn-s3-demo-bucket` 替换为您的存储桶的名称。

有关将 Amazon S3 存储桶策略添加到存储桶的更多信息，请参阅如何添加 S3 存储桶策略？ 有关批量工作流程的更多信息，请参阅获取批量物品推荐或获取批量用户细分。

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/*"
            ]
        }
    ]
}
```

用于导出数据集的 Amazon S3 存储桶策略

要导出数据集，Amazon Personalize 需要权限来针对 Amazon S3 存储桶使用 `PutObject` 和 `ListBucket` 操作。以下示例策略向 Amazon Personalize 主体授予 `PutObject` 和 `ListBucket` 权限。将 `amzn-s3-demo-bucket` 替换为您的存储桶名称，并将策略附加到存储桶。有关将 Amazon S3 存储桶策略添加到存储桶的信息，请参阅《Amazon Simple Storage Service 用户指南》中的如何添加 S3 存储桶策略？。

```
{
    "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/*"
            ]
        }
    ]
}
```

授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限

如果您在使用 Amazon Personalize 控制台时指定 AWS Key Management Service (AWS KMS) 密钥 APIs，或者如果您使用 AWS KMS 密钥加密 Amazon S3 存储桶，则必须授予亚马逊个性化使用您的密钥的权限。要授予权限，您的 AWS KMS 密钥策略和附加到您的服务角色的 IAM 策略必须授予 Amazon Personalize 使用您的密钥的权限。这适用于在 Amazon Personalize 中创建以下内容。

- 数据集组
- 数据集导入任务（只有 AWS KMS 密钥策略必须授予权限）
- 数据集导出作业
- 批量推理作业
- 批量细分作业
- 指标归因

您的 AWS KMS 密钥策略和 IAM 策略必须授予以下操作的权限：

- Decrypt
- GenerateDataKey
- DescribeKey
- CreateGrant（仅在密钥策略中为必填项）
- ListGrants

在创建资源后撤消 AWS KMS 密钥权限可能会导致在创建筛选条件或获取推荐时出现问题。有关 AWS KMS 策略的更多信息，请参阅AWS Key Management Service 开发人员指南中的在 AWS KMS 中使用密钥策略。有关创建 IAM 策略的信息，请参阅《IAM 用户指南》中的创建 IAM 策略。有关将 IAM 策略附加到角色的信息，请参阅《IAM 用户指南》中的添加和删除 IAM 身份权限。

主题

- 密钥策略示例
- IAM 策略示例

密钥策略示例

以下密钥策略示例向 Amazon Personalize 和角色授予执行前面的 Amazon Personalize 操作的最低权限。如果您在创建数据集组时指定了密钥并希望从数据集导出数据，则您的密钥策略必须包含 `GenerateDataKeyWithoutPlaintext` 操作。

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": "*"
        }
    ]
}
```

IAM 策略示例

以下 IAM 策略示例向角色授予之前的 Amazon Personalize 操作所需的最低 AWS KMS 权限。对于数据集导入任务，只有 AWS KMS 密钥策略需要授予权限。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey",
                "kms:ListGrants"
            ],
            "Resource": "*"
        }
    ]
}
```

设置 AWS CLI

AWS Command Line Interface (AWS CLI) 是一款用于管理 AWS 服务（包括 Amazon Personalize）的统一开发者工具。我们建议您安装它。

1. 要安装 AWS CLI，请按照《AWS Command Line Interface 界面用户指南》 AWS Command Line Interface中的 “安装” 中的说明进行操作。
2. 要配置 AWS CLI 和设置配置文件以调用 AWS CLI，请按照AWS Command Line Interface 用户指南中配置 AWS CLI中的说明进行操作。
3. 要确认配置 AWS CLI 文件配置正确，请运行以下命令。

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

   如果您的配置文件已正确配置，您将看到类似于以下内容的输出。

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

4. 要验证是否已配置 AWS CLI 为与 Amazon Personalize 配合使用，请运行以下命令。

   ```
   aws personalize help
   ```

   and

```
aws personalize-runtime help
```

和

```
aws personalize-events help
```

如果配置 AWS CLI 正确，您将看到 Amazon Personalize、Amazon Personalize 运行时和亚马逊个性化事件支持的 AWS CLI 命令列表。

如果您设置了， AWS CLI 但它无法识别 Amazon Personalize 的命令，请更新 AWS CLI。要更新 AWS CLI，请运行以下命令。

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

有关更多信息，请参阅使用 pip 安装 AWS CLI。

设置 AWS 软件开发工具包

下载并安装您要使用的 AWS 软件开发工具包。本指南提供了适用于 适用于 Python (Boto3) 的 AWS SDK Java 2.x 的 SDK 和适用于 JavaScript v3 的 SDK 的示例。有关其他 AWS 软件开发工具包的信息，请参阅适用于 Amazon Web Services 的工具。有关设置 Amplify 的信息，请参阅 Amplify 文档。

- 适用于 Python (Boto3) 的 AWS SDK

  要安装 SDK for Python (Boto3)，请按照 Boto3 文档中的快速入门说明进行操作。

- 适用于 Java 2.x 的 SDK

  要了解如何设置 SDK for Java 2.x，请参阅《AWS SDK for Java 2.x 开发人员指南》中的 SDK for Java 2.x 入门主题。

  有关 Amazon Personalize 的代码示例，请参阅 AWS SDK 示例存储库中的 Amazon Personalize Java 代码示例。

- 适用于 JavaScript 的 AWS SDK v3

  要了解如何为 JavaScript v3 设置 SDK，请参阅适用于 JavaScript 的 AWS SDK 开发者指南中的入门 适用于 JavaScript 的 AWS SDK主题。

有关 Amazon Personalize 的代码示例，请参阅软件开发工具包示例存储库中适用于 JavaScript v3 的软件开发工具包的 Amazon Per AWS sonalize 代码示例。

入门教程

下面几个部分有助于您通过 Amazon Personalize 控制台、AWS CLI 和 AWS SDK 开始使用 Amazon Personalize。教程使用包含来自 600 个用户对 9700 部电影的 10 万个电影评级的历史数据。

为了简化教程：

- 我们使用小数据集。这可能会对资源生成的任何指标产生负面影响。这些教程是对 Amazon Personalize 工作流程的介绍，不一定会生成性能极佳的模型。
- 我们仅创建物品交互数据集，依赖用户看过电影这个事实而非用户对电影的评级。这简化了训练数据的准备。
- 我们不记录实时用户交互事件。有关捕获用户事件的信息，请参阅记录实时事件以影响推荐。

您可以选择开始使用域数据集组或自定义数据集组：

- 域数据集组提供的资源根据您的域针对不同的使用案例进行了优化。要开始创建域数据集组，请完成入门先决条件，然后完成域数据集组入门中的教程。
- 自定义数据集组仅允许您创建和配置自定义资源。要开始使用自定义资源和 User-Personalization-v2 配方 配方为用户提供个性化影片推荐，请完成入门先决条件，然后开始自定义数据集组入门中的教程。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 入门先决条件
- 域数据集组入门
- 自定义数据集组入门

入门先决条件

以下步骤是入门练习的先决条件。

1. 设置权限，让 Amazon Personalize 可以代表您访问您的资源。这包括为 Amazon Personalize 创建服务角色，并通过 IAM 策略向其授予访问 Amazon Personalize 资源的权限。有关更多信息，请参阅 向 Amazon Personalize 授予访问资源的权限。
2. 准备您的训练数据并将该数据上传到 Amazon S3 存储桶：
   - 有关域数据集组教程，请参阅创建训练数据（域数据集组）。
   - 有关自定义数据集组教程，请参阅创建训练数据（自定义数据集组）。
3. 向您的 Amazon Personalize 服务角色授予访问 Amazon S3 资源的权限，如向 Amazon Personalize 授予访问 Amazon S3 资源的权限中所述。

创建训练数据（域数据集组）

要创建训练数据，请下载、修改电影评级数据，并将此数据保存到 Amazon Simple Storage Service (Amazon S3) 存储桶。然后，向 Amazon Personalize 提供从存储桶进行读取的权限。

创建训练数据

1. 下载并解压缩来自 MovieLens 针对教育和发展推荐 部分下的 ml-latest-small.zip 电影评级 zip 文件（作者：F. Maxwell Harper 和 Joseph A. Konstan，2015 年。The MovieLens Datasets: History and Context。ACM Transactions on Interactive Intelligent Systems (TiiS) 第 5 卷第 4 期：19:1–19:19。https://doi.org/10.1145/2827872）。
2. 打开 `ratings.csv`文件。此文件包含本教程的交互数据。
   a. 删除评级 列。
   b. 将 `userId` 和 `movieId` 列分别重命名为 `USER_ID` 和 `ITEM_ID`。
   c. 添加一个 EVENT_TYPE 列，并将每条记录的值设置为 `watch`。如果使用的是 Microsoft Excel，则您可以为每条记录设置 EVENT_TYPE，方法是：在列的第一个单元格输入 `watch`，然后双击该单元格的右下角。您的标题应如下所示：

      `USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE`

      这些列必须与所示内容完全相同，Amazon Personalize 才能识别数据。数据的前几行应如下所示：

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

保存 ratings.csv 文件。

3. 将 ratings.csv 上传到 Amazon S3 存储桶。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
4. 向 Amazon Personalize 提供在存储桶中读取数据的权限。有关更多信息，请参阅 向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

创建训练数据（自定义数据集组）

要创建训练数据，请下载、修改电影评级数据，并将此数据保存到 Amazon Simple Storage Service (Amazon S3) 存储桶。然后，向 Amazon Personalize 提供从存储桶进行读取的权限。

1. 下载并解压缩来自 MovieLens 针对教育和发展推荐 部分下的 ml-latest-small.zip 电影评级 zip 文件（作者：F. Maxwell Harper 和 Joseph A. Konstan，2015 年。The MovieLens Datasets: History and Context。ACM Transactions on Interactive Intelligent Systems (TiiS) 第 5 卷第 4 期：19:1–19:19。https://doi.org/10.1145/2827872）。
2. 打开 ratings.csv文件。此文件包含本教程的交互数据。

   a. 删除评级 列。

   b. 使用以下内容替换标题行：

      USER_ID,ITEM_ID,TIMESTAMP

      这些标题必须与所示内容完全相同，Amazon Personalize 才能识别数据。

   保存 ratings.csv 文件。
3. 将 ratings.csv 上传到 Amazon S3 存储桶。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
4. 向 Amazon Personalize 提供在存储桶中读取数据的权限。有关更多信息，请参阅 向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

域数据集组入门

在本入门教程中，您将为 VIDEO_ON_DEMAND 域创建域数据集组，从 CSV 文件导入交互数据，并为热门精选 使用案例创建推荐器。然后，您将使用推荐器，为用户获取个性化电影推荐。教程使用包含来自 600 个用户对 9700 部电影的 10 万个电影评级的历史数据。

首先，完成入门先决条件，然后根据您想如何创建 Amazon Personalize 资源，继续执行域数据集组入门（控制台）、域数据集组入门 (SDK for Python (Boto3))、域数据集组入门 (SDK for Java 2.x)或域数据集组入门 (SDK for JavaScript v3)。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 域数据集组入门（控制台）
- 域数据集组入门 (SDK for Java 2.x)
- 域数据集组入门 (SDK for Python (Boto3))
- 域数据集组入门 (SDK for JavaScript v3)

域数据集组入门（控制台）

在本练习中，您将使用 Amazon Personalize 控制台创建域数据集组和推荐器，以便为给定用户返回电影推荐。

在开始本练习之前，请查看入门先决条件。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

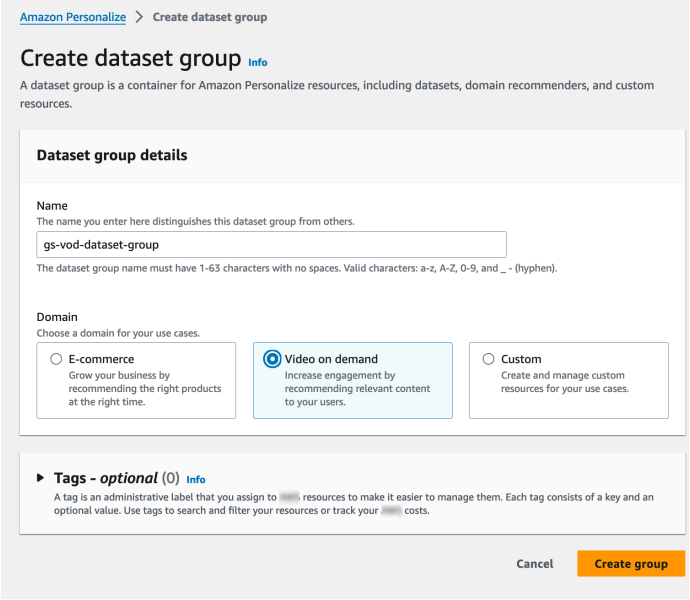
步骤 1：创建域数据集组

在此过程中，您将为 VIDEO_ON_DEMAND 域创建域数据集组，使用 VIDEO_ON_DEMAND 域的默认架构创建物品交互数据集，然后导入您在创建训练数据（域数据集组）中创建的物品交互数据。

创建域数据集组

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。

2. 在导航窗格中，选择创建数据集组。
3. 在数据集组详细信息中，为您的数据集组指定一个名称。
4. 对于域，选择点播视频。您选择的域决定了您在导入数据时使用的默认架构。它还决定了哪些使用案例可供推荐器使用。您的屏幕应类似于以下内容。



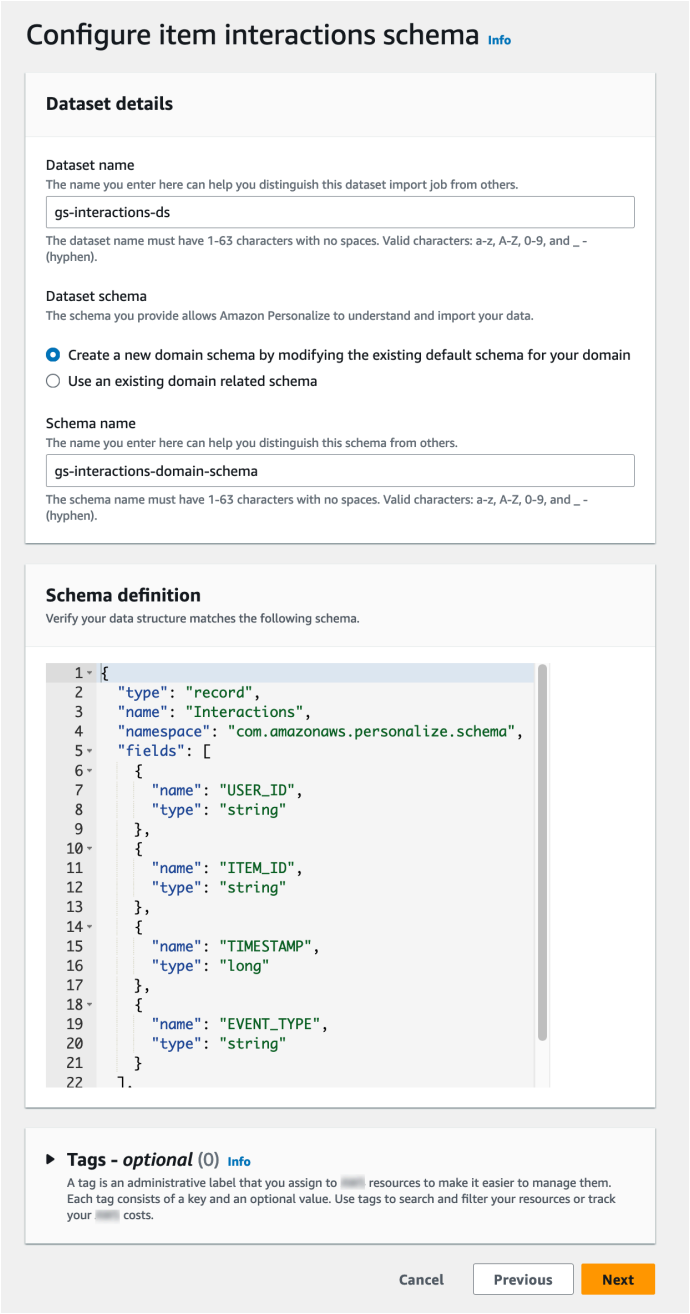
5. 选择创建数据集组。这时将显示“概览”页面。继续执行步骤 2：导入数据。

步骤 2：导入数据

在此过程中，您将使用默认 VIDEO_ON_DEMAND 域架构创建物品交互数据集。然后，导入您在创建训练数据（域数据集组）中创建的物品交互数据。

导入数据

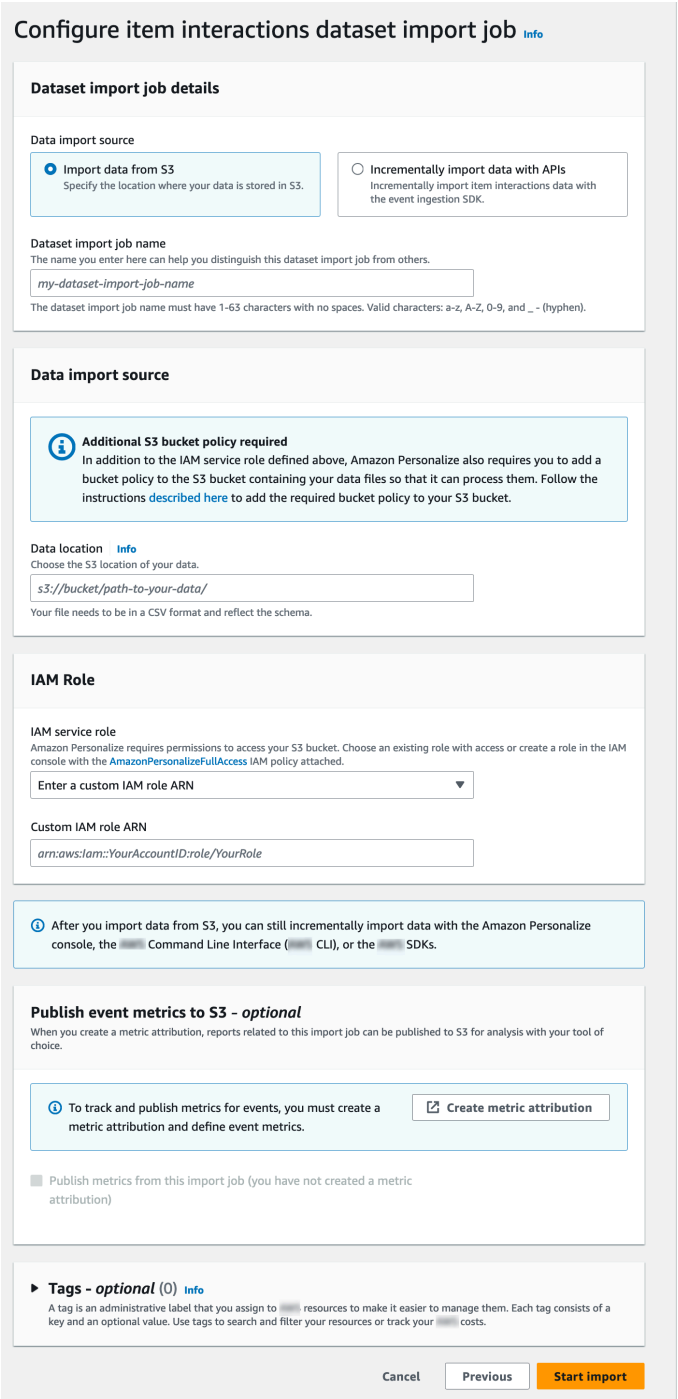
1. 在概述页面上，对于步骤 1. 创建数据集并导入数据，选择创建数据集，然后选择物品交互数据集。
2. 选择将数据直接导入 Amazon Personalize 数据集，然后选择下一步。
3. 在配置物品交互架构页面上，对于数据集名称，提供物品交互数据集的名称。
4. 对于数据集架构，选择通过修改域的现有默认架构来创建新域架构，然后输入架构的名称。架构定义将更新，以显示 VIDEO_ON_DEMAND 域的默认架构。保持架构不变。您的屏幕应类似于以下内容。



5. 选择下一步。将出现配置物品交互数据集导入作业页面。
6. 在配置物品交互数据集导入作业页面上，将数据导入源保留为从 S3 导入数据不变。
7. 对于数据集导入作业名称，为导入作业提供名称。
8. 对于数据导入源，指定 Amazon Simple Storage Service（S3）中存储数据的位置。使用以下语法：

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

9. 在 IAM 角色中，对于 IAM 服务角色，选择输入自定义 IAM 角色 ARN，然后输入您在为 Amazon Personalize 创建 IAM 角色中创建的角色的 Amazon 资源名称 (ARN)。您的屏幕应类似于以下内容。



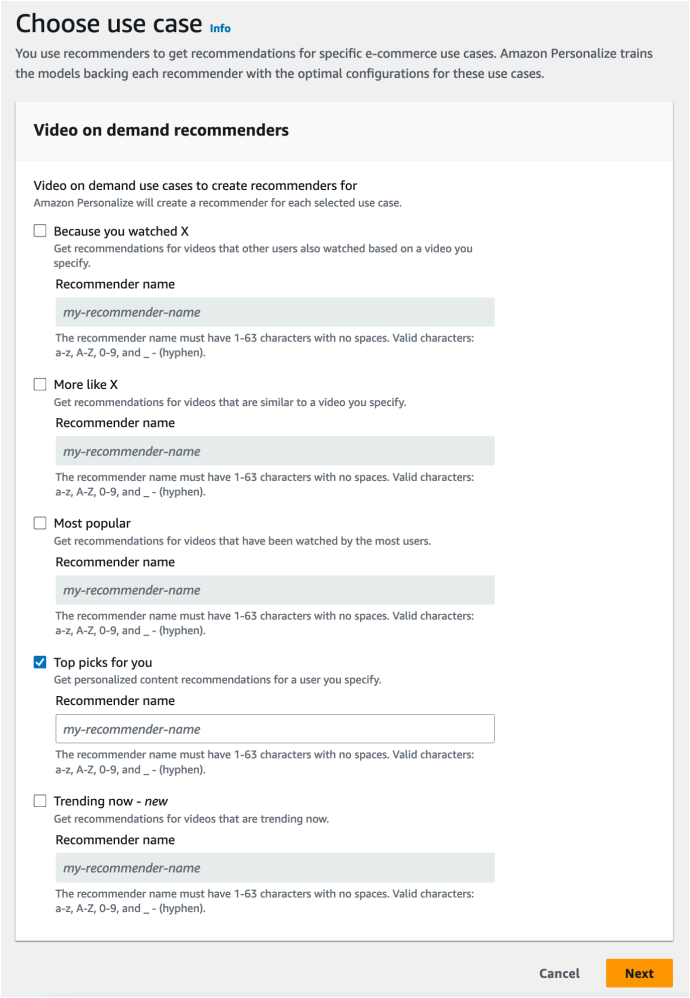
10. 选择开始导入以导入数据。随后，会显示域数据集组的概览页面。注意设置数据集部分中的导入状态。当状态为 `Interaction data active` 时，继续步骤 3：创建推荐器。

步骤 3：创建推荐器

在此过程中，您将为 VIDEO_ON_DEMAND 域的热门精选 使用案例创建推荐器。

创建推荐器

1. 在域数据集组的概述页面上，在步骤 3 中选择使用视频点播推荐器选项卡，然后选择创建推荐器。
2. 在选择应用场景页面上，选择热门精选，并提供推荐器名称。您的屏幕应显示以下内容。



3. 选择下一步。
4. 使高级配置页面上的字段保持不变，然后选择下一步。
5. 查看推荐器详细信息并选择创建推荐器来创建您的推荐器。

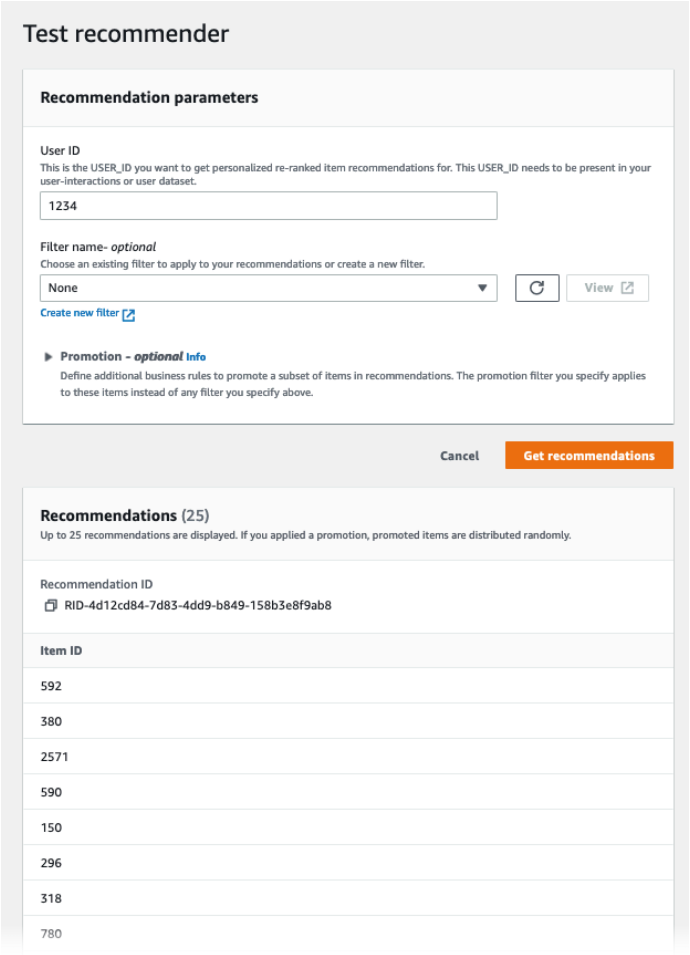
   您可以在推荐器页面上，监控每个推荐器的状态。当推荐器状态为活动时，您可以在步骤 4：获得建议中使用它来获取建议。

步骤 4：获得建议

在此过程中，使用您在前一个步骤中创建的推荐器来获得建议。

获得建议

1. 在域数据集组的“概览”页面上，在导航窗格中选择推荐器。
2. 在推荐器页面上，选择您的推荐器。
3. 在右上角，选择测试。
4. 在推荐器参数 中，输入用户 ID。使其他字段保持不变。
5. 选择获得建议。此时，将显示一个表格，其中包含向用户推荐的前 25 个物品。您的屏幕应类似于以下内容。



域数据集组入门 (SDK for Java 2.x)

本教程向您展示如何使用 SDK for Java 2.x 为 VIDEO_ON_DEMAND 域创建域数据集组。在本教程中，您将为热门精选 使用案例创建推荐器。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

先决条件

以下是完成本教程的先决条件步骤：

- 完成入门先决条件，设置所需权限并创建训练数据。如果您还完成了域数据集组入门（控制台），则可以重复使用相同的源数据。如果您使用自己的源数据，请确保按照先决条件中的步骤设置数据格式。
- 设置 SDK for Java 2.x 环境和 AWS 凭证，如《AWS SDK for Java 2.x 开发人员指南》中的设置 AWS SDK for Java 2.x 过程所述。

教程

在以下步骤中，您将项目设置为使用 Amazon Personalize 程序包并创建 Amazon Personalize SDK for Java 2.x 客户端。然后，导入数据，为热门精选 使用案例创建推荐器，并获得建议。

步骤 1：设置您的项目以使用 Amazon Personalize 程序包

完成先决条件后，将 Amazon Personalize 依赖项添加到 pom.xml 文件中，然后导入 Amazon Personalize 程序包。

1. 将以下依赖项添加到 pom.xml 文件中。最新版本号可能与示例代码不同。

```
<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. 将以下导入语句添加至您的项目。

```
// 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;
```

步骤 2：创建 Amazon Personalize 客户端

将 Amazon Personalize 依赖项添加到 pom.xml 文件并导入所需的程序包之后，创建以下 Amazon Personalize 客户端：

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

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

步骤 3：导入数据

初始化 Amazon Personalize 客户端后，导入您在完成入门先决条件时创建的历史数据。要将历史数据导入 Amazon Personalize，请执行以下操作：

1. 将以下 Avro 架构作为 JSON 文件保存到您的工作目录中。此架构与您在完成创建训练数据（域数据集组）时创建的 CSV 文件中的列相匹配。

```
{
  "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. 使用以下 createDomainSchema 方法创建 Amazon Personalize 域架构。将以下内容作为参数传递：Amazon Personalize 服务客户端、架构名称、域的 VIDEO_ON_DEMAND，以及您在上一步中创建的架构 JSON 文件的文件路径。该方法将返回新架构的 Amazon 资源名称 (ARN)。请将其存储以便将来使用。

```
    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. 创建数据集组。使用以下 createDomainDatasetGroup 方法创建域数据集组。将以下内容作为参数传递：Amazon Personalize 服务客户端、数据集组的名称以及域的传递 VIDEO_ON_DEMAND。该方法将返回新数据集组的 ARN。请将其存储以便将来使用。

```
   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. 创建物品交互数据集。使用以下 `createDataset` 方法创建物品交互数据集。将以下内容作为参数传递：Amazon Personalize 服务客户端、数据集名称、架构的 ARN、数据集组的 ARN 以及数据集类型的 `Interactions`。该方法将返回新数据集的 ARN。请将其存储以便将来使用。

```
   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. 使用数据集导入作业导入数据。使用以下 `createPersonalizeDatasetImportJob` 方法创建数据集导入作业。

   将以下内容作为参数传递：Amazon Personalize 服务客户端、作业名称和交互数据集的 ARN。传递存储训练数据的 Amazon S3 存储桶路径 (s3://*bucket name*/*folder name*/ratings.csv) 以及服务角色的 ARN。您在入门先决条件期间创建了此角色。该方法将返回数据集导入作业的 ARN。（可选）将其存储起来以备后用。

```
    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 "";
}
```

步骤 4：创建推荐器

数据集导入作业完成后，您就可以创建推荐器了。使用以下 `createRecommender` 方法创建推荐器。将以下内容作为参数传递：Amazon Personalize 服务客户端、推荐器的名称、数据集组的 Amazon 资源名称 (ARN) 以及食谱 ARN 的 `arn:aws:personalize:::recipe/aws-vod-top-picks`。该方法将返回新推荐器的 ARN。请将其存储以便将来使用。

```
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 "";
    }
```

步骤 5：获得建议

创建推荐器后，您可以使用它来获取建议。使用以下 getRecs 方法为用户获取建议。将以下内容作为参数传递：Amazon Personalize 运行时客户端、您在上一步中创建的推荐器的 Amazon 资源名称 (ARN) 和用户 ID（例如 123）。该方法会在屏幕上打印出推荐物品列表。

```
    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);
        }
    }
```

域数据集组入门 (SDK for Python (Boto3))

本教程向您展示如何使用 SDK for Python (Boto3) 为 VIDEO_ON_DEMAND 域创建域数据集组。在本教程中，您将为热门精选 使用案例创建推荐器。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 先决条件
- 教程
- 开始使用 Jupyter（iPython）APIs 笔记本电脑的 Amazon Personalize

先决条件

以下是使用本指南中的 Python 示例的先决条件步骤：

- 完成入门先决条件，设置所需权限并创建训练数据。如果您使用自己的源数据，请确保按照先决条件中的步骤设置数据格式。
- 按照中的说明设置您的 适用于 Python (Boto3) 的 AWS SDK 环境设置 AWS 软件开发工具包。

教程

在以下步骤中，您将验证您的环境并为 Amazon Personalize 创建 SDK for Python (Boto3) 客户端。然后，导入数据，为热门精选 使用案例创建推荐器，并获得建议。

步骤 1：验证您的 Python 环境并创建 boto3 客户端

在完成先决条件之后，请运行以下 Python 示例，以确认您的环境已正确配置。此代码还会创建您在本教程中使用的 Amazon Personalize boto3 客户端。如果已正确配置环境，则会显示可用食谱列表，您现在可以运行本教程中的其他示例。

```
import boto3

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

response = personalize.list_recipes()
```

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

步骤 2：导入数据

创建 Amazon Personalize boto3 客户端并验证您的环境后，导入您在完成入门先决条件时创建的历史数据。要将历史数据导入 Amazon Personalize，请执行以下操作：

1. 使用以下代码创建 Amazon Personalize 架构。将 gs-domain-interactions-schema 替换为架构的名称。

```
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. 使用以下代码创建数据集组。将 dataset group name 替换为数据集组的名称。

```
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. 使用以下代码在新数据集组中创建物品交互数据集。为数据集命名，并提供前面步骤中的schema_arn和dataset_group_arn。

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

dataset_arn = response['datasetArn']
```

4. 使用以下代码通过数据集导入作业导入数据。代码使用 describe_dataset_import_job 方法来跟踪作业的状态。

   将以下内容作为参数传递：作业名称、上一步中的 dataset_arn、存储训练数据的 Amazon S3 存储桶路径 (s3://*bucket name*/*folder name*/ratings.csv) 以及您的 IAM 服务角色的 ARN。您在入门先决条件期间创建了此角色。Amazon Personalize 需要权限才能访问存储桶。有关授予访问权限的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

```
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)
```

步骤 4：创建推荐器

数据集导入作业完成后，您就可以创建推荐器了。使用以下代码创建推荐器。将以下内容作为参数传递：推荐器的名称、数据集组的 Amazon 资源名称 (ARN) 以及食谱 ARN 的 `arn:aws:personalize:::recipe/aws-vod-top-picks`。代码使用 describe_recommender 方法来跟踪推荐器的状态。

```
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)
```

步骤 5：获得建议

创建推荐器后，您可以使用它，通过以下代码获取建议。将您在上一步中创建的推荐器的 Amazon 资源名称 (ARN) 和用户 ID（例如 123）作为参数传递。方法会打印推荐物品列表。

```
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'])
```

开始使用 Jupyter（iPython） APIs 笔记本电脑的 Amazon Personalize

要开始使用 Jupyter 笔记本创建域数据集组，请克隆或下载在 Amazon Personalize 示例存储库的 notebooks_managed_domains 文件夹中找到的一系列笔记本。笔记本将引导您使用 Amazon Personalize 导入训练数据、创建推荐器以及获取建议。

> **Note**
>
> 在开始使用笔记本之前，请务必按照 README.md 中的步骤构建您的环境

域数据集组入门 (SDK for JavaScript v3)

本教程向您展示如何使用 适用于 JavaScript 的 AWS SDK v3 为 VIDEO_ON_DEMAND 域创建域数据集组。在本教程中，您将为热门精选 使用案例创建推荐器。

要在 GitHub 上查看本教程中使用的代码，请参阅《AWS SDK 代码示例》存储库中的 Amazon Personalize SDK for JavaScript v3 代码示例。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 先决条件
- 教程

先决条件

以下是完成本教程的先决条件步骤：

- 完成入门先决条件，设置所需权限并创建训练数据。如果您还完成了域数据集组入门（控制台），则可以重复使用相同的源数据。如果您使用自己的源数据，请确保按照先决条件中的步骤设置数据格式。
- 设置 SDK for JavaScript 和 AWS 凭证，如《适用于 JavaScript 的 AWS SDK 开发人员指南》中的设置 SDK for JavaScript 过程所述。

教程

在以下步骤中，您将安装所需的依赖项。然后，创建一个数据集组，导入数据，为热门精选 使用案例创建推荐器，并获得建议。

如果您使用 Node.js，则可以通过将示例保存为 JavaScript 文件然后运行 `node <fileName.js>`，运行每个代码示例。

步骤 1：安装 Amazon Personalize 依赖项

完成先决条件后，安装以下 Amazon Personalize 依赖项：

- @aws-sdk/client-personalize
- @aws-sdk/client-personalize-runtime
- @aws-sdk/client-personalize-events（在本教程中可选，但如果您想在创建推荐器之后记录事件，则为必需项）

以下是您可以使用的 package.json 文件示例。要使用 Node.js 安装依赖项，请导航到保存 package.json 文件的地方并运行 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
  }
}
```

步骤 2：创建 Amazon Personalize 客户端

安装依赖项后，创建您的 Amazon Personalize 客户端。在本教程中，代码示例假设您在名为 personalizeClients.js 的文件（存储在名为 libs 的目录中）中创建客户端。

以下是文件 personalizeClient.js 的示例。

```
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 };
```

步骤 3：导入数据

创建 Amazon Personalize 客户端后，导入您在完成入门先决条件时创建的历史数据。要将历史数据导入 Amazon Personalize，请执行以下操作：

1. 将以下 Avro 架构作为 JSON 文件保存到您的工作目录中。此架构与您在完成创建训练数据（域数据集组）时创建的 CSV 文件中的列相匹配。

```
{
  "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. 使用以下 createDomainSchema.js 代码创建 Amazon Personalize 域架构。将 SCHEMA_PATH 替换为您刚创建的 schema.json 文件的路径。更新 createSchemaParam 以指定架构的名称，对于 domain，指定 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. 使用以下 createDomainDatasetGroup.js 代码创建 Amazon Personalize 域数据集组。更新 domainDatasetGroupParams 以指定数据集组的名称，对于 domain，指定 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. 使用以下 createDataset.js 代码在 Amazon Personalize 中创建物品交互数据集。更新 createDatasetParam，以指定您刚创建的数据集组和架构的 Amazon 资源名称 (ARN)，为数据集命名，并对于 datasetType，指定 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. 使用以下 `createDatasetImportJob.js` 代码导入数据。更新 `datasetImportJobParam` 以指定以下内容：
   - 为作业指定名称并指定交互数据集的 ARN。
   - 对于 `dataLocation`，指定您存储训练数据的 Amazon S3 存储桶路径 (s3://*https://amzn-s3-demo-bucket.s3.region-code.amazonaws.com*/*folder name*/ratings.csv)。
   - 对于 `roleArn`，为 Amazon Personalize 服务角色指定 Amazon 资源名称。您在入门先决条件期间创建了此角色。

```
// 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();
```

步骤 4：创建推荐器

数据集导入作业完成后，您就可以创建推荐器了。要创建推荐器，请使用以下 createRecommender.js 代码。使用以下内容更新 createRecommenderParam：为推荐器指定名称，指定数据集组的 ARN，然后对于 recipeArn，指定 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();
```

步骤 5：获得建议

创建推荐器后，您可以使用它来获取建议。使用以下 getRecommendations.js 代码为用户获取建议。更新 getRecommendationsParam，以指定您在上一步中创建的推荐器的 ARN，并指定用户 ID（例如 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();
```

自定义数据集组入门

> **Important**
>
> 在本教程中，创建一个使用自动训练的解决方案。默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，请务必在结束后删除解决方案。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

本入门指南介绍了如何使用自定义数据集组和 User-Personalization-v2 配方 配方为用户提供个性化电影推荐。教程使用包含来自 600 个用户对 9700 部电影的 10 万个电影评级的历史数据。

首先，请完成入门先决条件，然后继续执行开始使用（控制台）、入门 (AWS CLI)、入门 (SDK for Python (Boto3))或入门 (SDK for Java 2.x)。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 开始使用（控制台）
- 入门 (AWS CLI)
- 入门 (SDK for Python (Boto3))
- 入门 (SDK for Java 2.x)

开始使用（控制台）

在本练习中，您将使用 Amazon Personalize 控制台创建一个自定义数据集组，其中含有可为给定用户返回电影推荐的解决方案。在开始本练习之前，请查看入门先决条件。

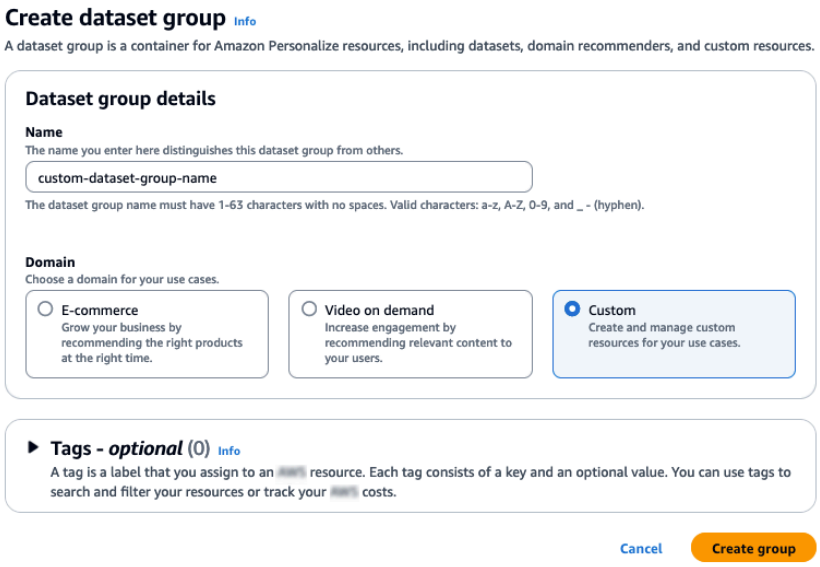
完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

步骤 1：创建数据集组和数据集

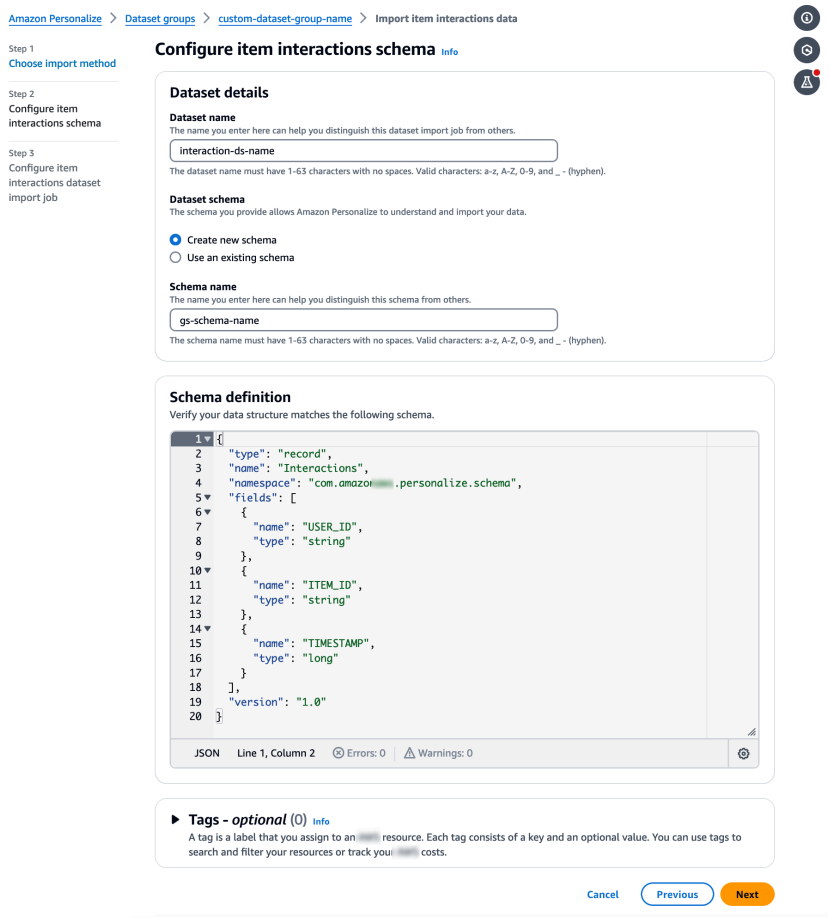
在此过程中，您首先创建一个数据集组。接下来，在数据集组中创建一个 Amazon Personalize 物品交互数据集。

创建数据集组和数据集

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择创建数据集组。
3. 在数据集组详细信息中，对于数据集组名称，为您的数据集组指定一个名称。
4. 对于域，选择自定义。您的屏幕应类似于以下内容：



5. 选择创建群组。这时将显示概览页面。
6. 在步骤 1. 创建数据集并导入数据，选择创建数据集，然后选择物品交互数据集。
7. 选择将数据直接导入 Amazon Personalize 数据集，然后选择下一步。
8. 在配置物品交互数据集页面上，对于数据集名称，为数据集指定名称。
9. 对于数据集架构，选择创建新架构。在架构定义部分，将显示一个最小物品交互架构。该架构与您之前添加到 `ratings.csv` 文件中的标题相匹配，因此您无需进行任何更改。如果您尚未创建训练数据，请参阅入门先决条件。
10. 对于架构名称，为新架构指定一个名称。您的屏幕应类似于以下内容：



11. 选择下一步。将出现配置物品交互数据集导入作业页面。接下来，完成步骤 2：导入物品交互数据，以导入交互数据。

步骤 2：导入物品交互数据

现在您已经创建了一个数据集，是时候将物品交互数据导入数据集了。

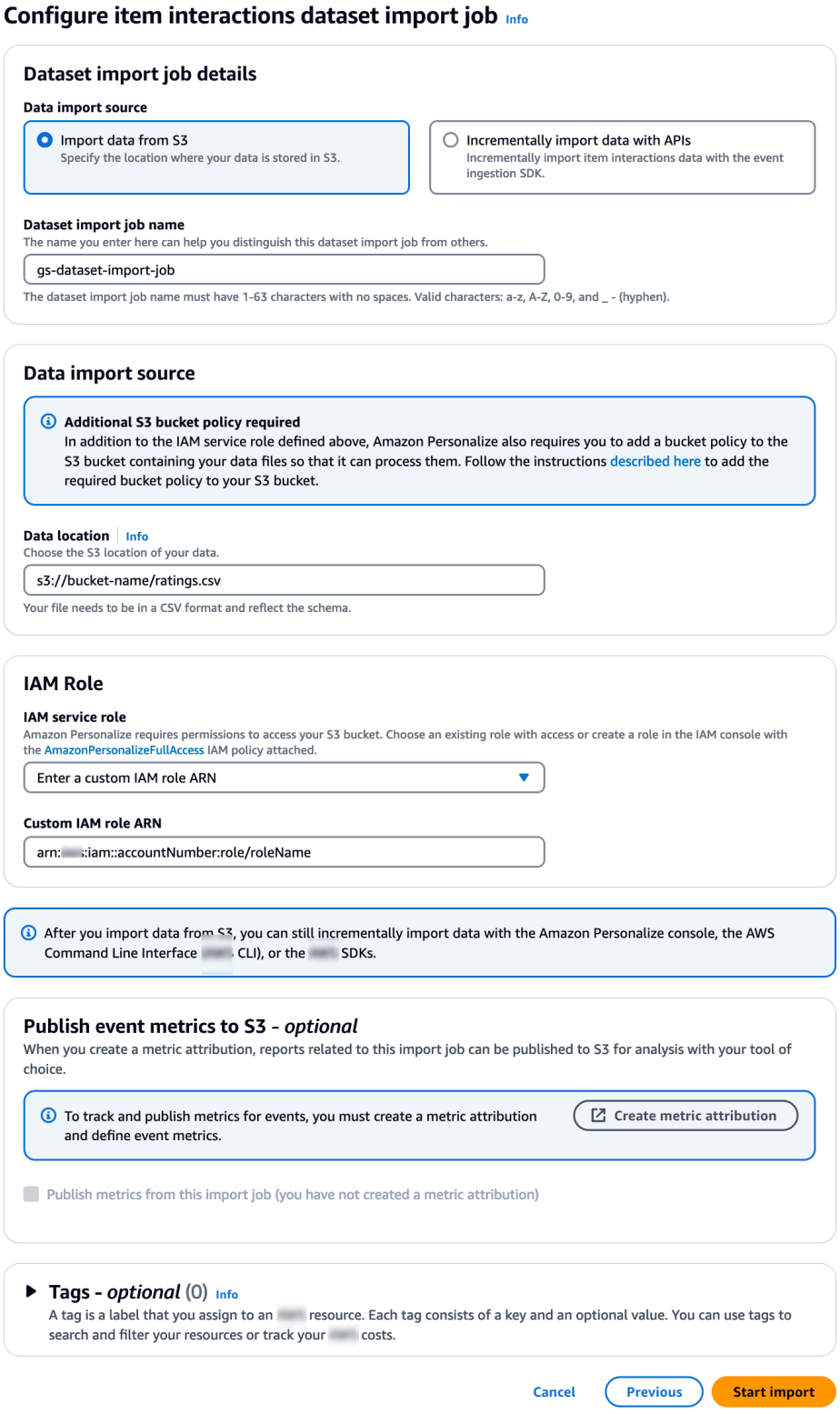
导入物品交互数据

1. 在配置物品交互数据集导入作业页面上，对于数据导入源，选择从 S3 导入数据。
2. 对于数据集导入作业名称，为导入作业指定名称。
3. 在需要其他 S3 存储桶策略对话框中，如果您尚未授予 Amazon Personalize 权限，请按照说明添加所需的 Amazon S3 存储桶策略。
4. 对于数据位置，指定 Amazon Simple Storage Service (S3) 中存储电影数据文件的位置。使用以下语法：

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

5. 在 IAM 角色部分，对于 IAM 服务角色，选择输入自定义 IAM 角色 ARN。
6. 对于自定义 IAM 角色 ARN，指定您在为 Amazon Personalize 创建 IAM 角色中创建的角色。

   数据集导入作业详细信息和 IAM 角色部分应类似于以下内容：



7. 将事件指标发布到 S3 和标签部分保持不变，然后选择开始导入。数据导入作业开始，并显示概览页面。最初，状态为创建待处理（然后是创建正在进行中），且创建解决方案按钮处于禁用状态。

当数据导入作业已完成时，状态将变为活动，并且创建解决方案按钮已启用。

8. 现在，您已经导入了数据，可以在步骤 3：创建解决方案中创建解决方案了。

步骤 3：创建解决方案

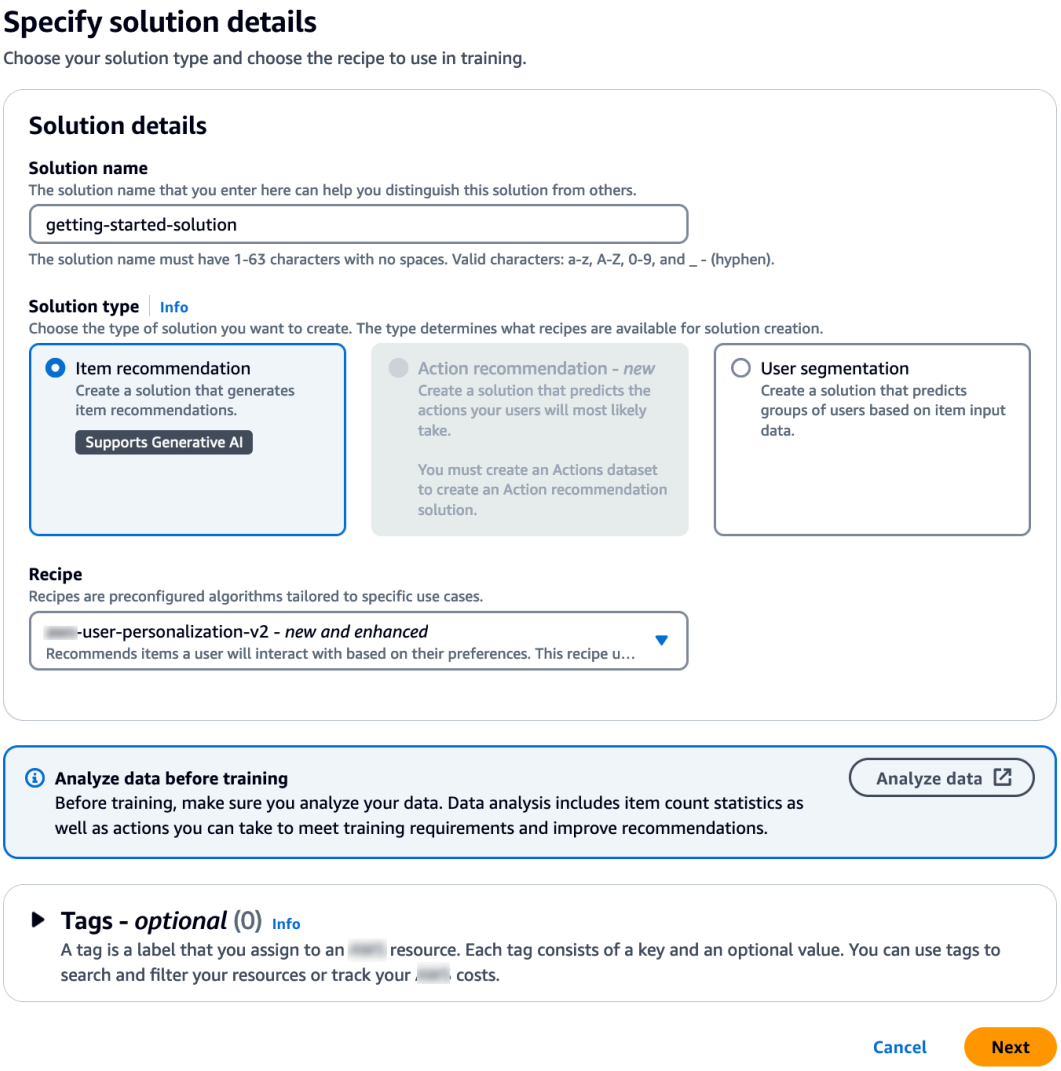
在本教程中，使用在步骤 2：导入物品交互数据中导入的数据集来训练模型。经过训练的模型称为解决方案版本。

> **Important**
>
> 在本教程中，创建一个使用自动训练的解决方案。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，请务必在结束后删除解决方案。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

创建解决方案

1. 在数据集组的概览页面上，在步骤 3. 设置训练和推荐资源中，选择创建解决方案。
2. 对于解决方案名称，为您的解决方案指定名称。
3. 对于解决方案类型，选择物品建议。
4. 在配方中，选择 `aws-user-personalization-v2`。

   您的屏幕应类似于以下内容：



5. 选择下一步。训练配置字段保持不变。您创建的解决方案每 7 天自动训练新模型，并对最新的物品交互数据给予更多权重。
6. 选择下一步，并查看解决方案的详细信息。
7. 选择创建解决方案，此时会显示解决方案的详细信息页面。创建解决方案后，Amazon Personalize 将在一小时内开始创建第一个解决方案版本。训练开始后，它会显示在详细信息页面的解决方案版本部分，并且您可以监控训练状态。

   当解决方案版本状态为活动 时，您就可以移动到步骤 4：创建市场活动了。

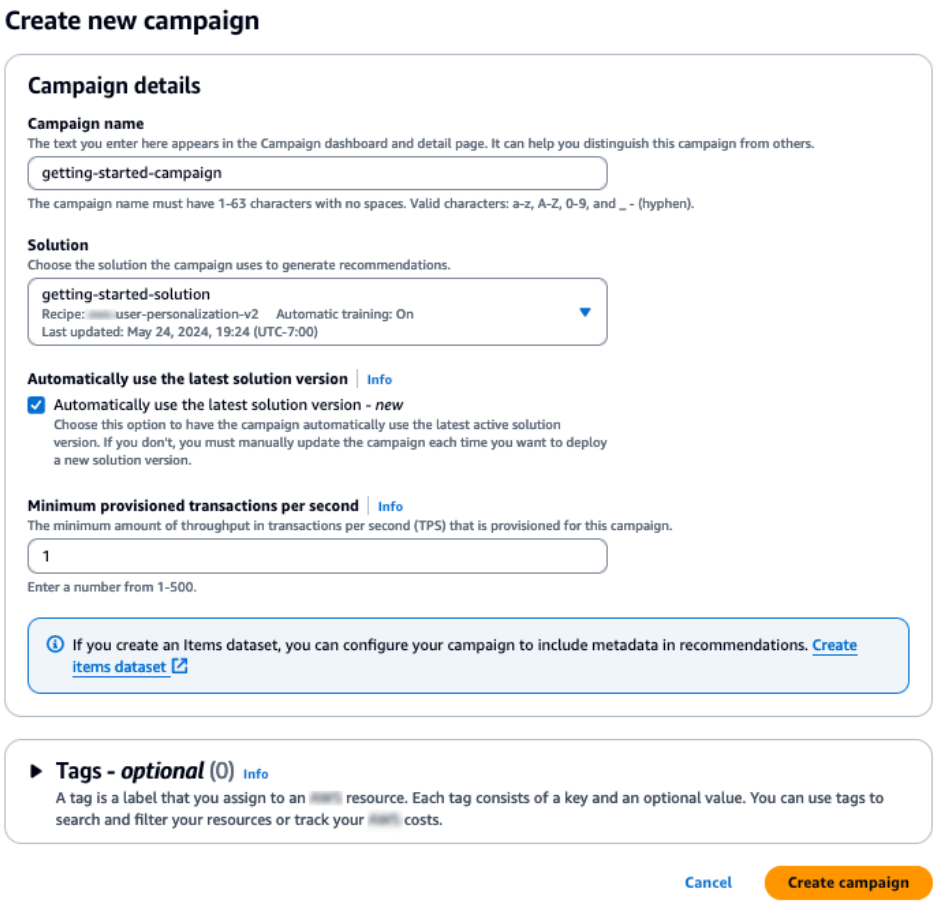
步骤 4：创建市场活动

在本过程中，您创建市场活动，以部署您在上一步中创建的解决方案版本。

创建市场活动

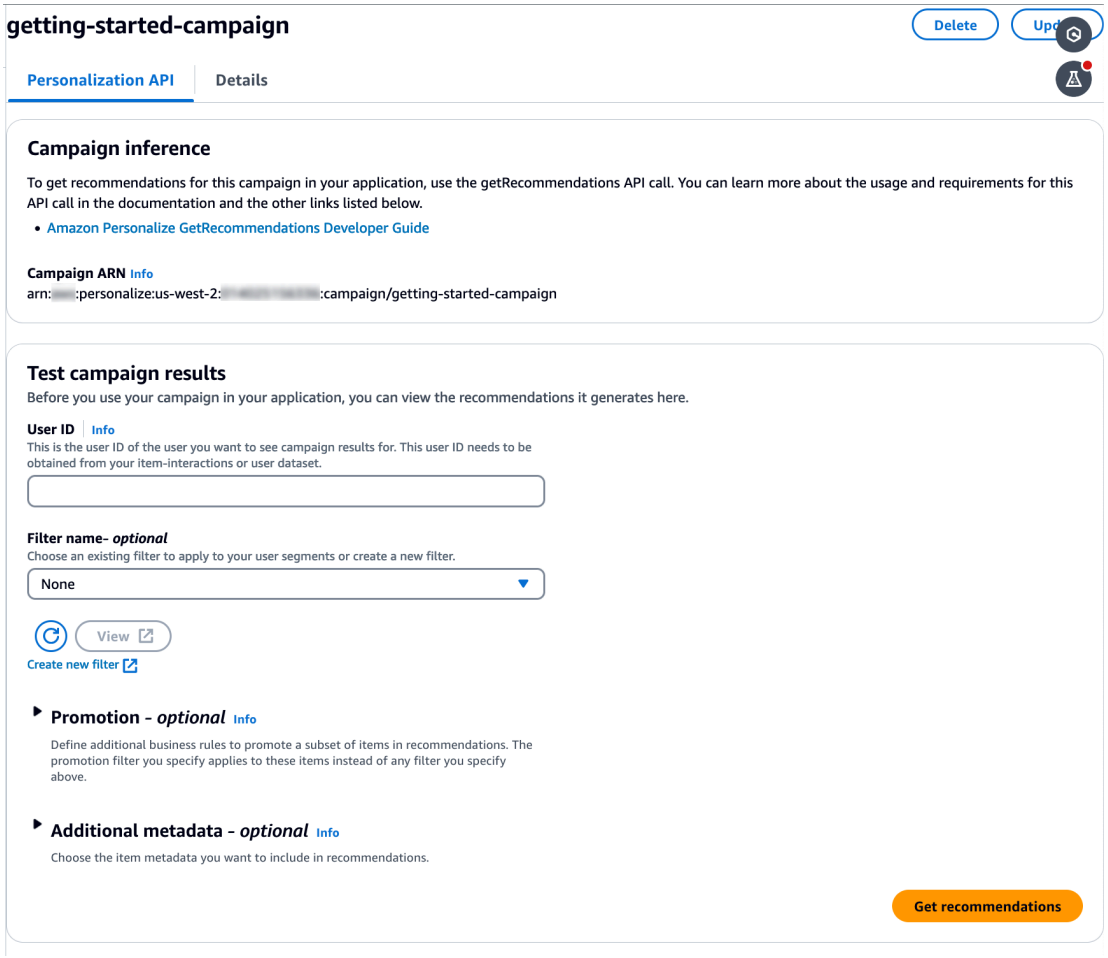
1. 在导航窗格中，展开自定义资源，然后选择市场活动。
2. 选择创建市场活动。此时，将显示创建新的市场活动页面。
3. 在市场活动详细信息中，对于市场活动名称，为您的市场活动指定一个名称。
4. 对于解决方案，选择您在上一步中创建的解决方案。
5. 选择自动使用最新的解决方案版本。所有其他字段保持不变。

   您的屏幕应类似于以下内容：



6. 选择创建市场活动。开始创建市场活动，并显示带有个性化 API 部分的市场活动详情页面。

   创建市场活动可能需要几分钟。Amazon Personalize 创建完市场活动后，页面将更新以显示测试市场活动结果部分。您的屏幕应类似于以下内容：



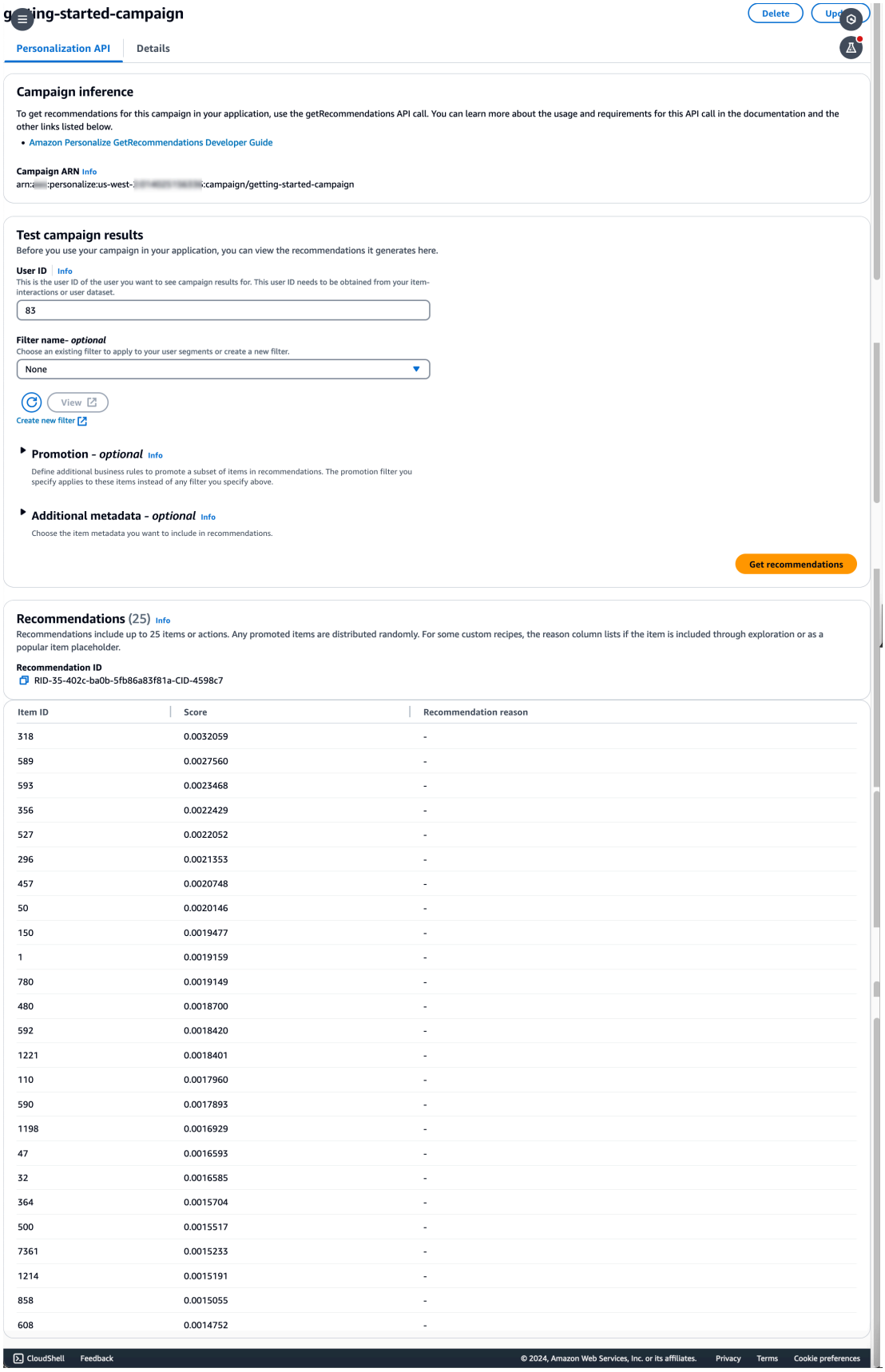
步骤 5：获得建议

在此过程中，使用您在前一个步骤中创建的活动来获得建议。

获得建议

1. 在测试市场活动结果中，对于用户 ID，指定评级 数据集中的值，例如，**83**。所有其他字段保持不变。
2. 选择获得建议。建议面板会列出建议物品的物品 ID 和分数。

   您的屏幕应类似于以下内容：


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

入门 (AWS CLI)

在本练习中，您将使用 AWS Command Line Interface (AWS CLI) 探索 Amazon Personalize。您创建一个活动，该活动返回给定用户 ID 的电影推荐。

在开始本练习之前，请先执行以下操作：

- 查看入门入门先决条件。
- 设置 AWS CLI，如设置 AWS CLI中所述。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

> **Note**
>
> 本练习中的 AWS CLI 命令在 Linux 上进行了测试。有关在 Windows 上使用 AWS CLI 命令的信息，请参阅《AWS Command Line Interface 用户指南》中的为 AWS Command Line Interface 指定参数值。

步骤 1：导入训练数据

按照步骤操作以创建一个数据集组，向此组添加数据集，然后使用电影评级数据填充此数据集。

1. 通过运行以下命令来创建数据集组。您可以通过传递 AWS Key Management Service 密钥 ARN 和对该密钥具有访问权限的 IAM 角色的 ARN 作为输入参数来加密数据集组。有关 API 的更多信息，请参阅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
   ```

   这将显示数据集组 ARN，例如：

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

使用 describe-dataset-group 命令显示您创建的数据集组，并指定返回的数据集组 ARN。

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

这将显示此数据集组及其属性，例如：

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

> **Note**
>
> 在数据集组中创建数据集之前，等待该组的 status 显示为“ACTIVE”。此操作通常很快。

如果您不记得数据集组 ARN，则使用 list-dataset-groups 命令显示您创建的所有数据集组及其 ARN。

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

> **Note**
>
> describe-object 和 list-objects 命令适用于大多数 Amazon Personalize 对象。这些命令没有在本练习的其余部分中显示，但它们是可用的。

2. 通过将以下代码保存到名为 MovieRatingSchema.json 的文件来创建一个 JSON 格式的架构文件。此架构与您之前添加到 ratings.csv 的标题匹配。架构名称为 Interactions，它与 Amazon Personalize 识别的数据集之一匹配。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

```
{
  "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. 通过运行以下命令创建架构。指定您在上一个步骤中保存的文件。此示例将该文件显示为属于当前文件夹。有关 API 的更多信息，请参阅CreateSchema。

```
aws personalize create-schema \
  --name MovieRatingSchema \
  --schema file://MovieRatingSchema.json
```

这将显示架构 Amazon 资源名称 (ARN)，例如：

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

4. 通过运行以下命令来创建一个空数据集。提供之前的步骤中返回的数据集组 ARN 和架构 ARN。`dataset-type` 必须与上一个步骤中的架构 `name` 匹配。有关 API 的更多信息，请参阅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
```

这将显示数据集 ARN，例如：

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

5. 将训练数据添加到数据集。

   a. 通过运行以下命令来创建数据集导入作业。提供之前步骤中返回的数据集 ARN 和 Amazon S3 存储桶名称。提供您在为 Amazon Personalize 创建 IAM 角色中创建的 AWS Identity and Access Management (IAM) 角色 ARN。有关 API 的更多信息，请参阅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
   ```

   这将显示数据集导入作业 ARN，例如：

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

   b. 通过使用 `describe-dataset-import-job` 命令来检查状态。提供上一个步骤中返回的数据集导入作业 ARN。有关 API 的更多信息，请参阅DescribeDatasetImportJob。

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

这将显示数据集导入作业的属性，包括其状态。最初，status 显示为 CREATE PENDING，例如：

```
{
  "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
  }
}
```

当状态显示为 ACTIVE 时，数据集导入完成。然后，您可以使用指定的数据集来训练模型。

> **Note**
>
> 导入需要耗费时间。等待数据集导入完成，然后使用数据集训练模型。

步骤 2：创建解决方案（训练模型）

要训练模型，您需要使用 CreateSolution 操作创建用于训练模型的配置，并开启自动训练。该解决方案会在一小时内自动开始训练第一个解决方案。

使用食谱和训练数据来训练模型。Amazon Personalize 提供一组预定义的食谱。有关更多信息，请参阅 选择食谱。在本练习中，您使用的是 User-Personalization-v2 配方。

1. 通过运行以下命令创建用于训练模型的配置。此命令创建一个使用自动训练的解决方案。该解决方案每七天（默认值）自动创建一个新的解决方案版本。

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

这将显示解决方案 ARN，例如：

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

2. 使用 `describe-solution` 命令检查创建 状态。提供上一步中返回的解决方案 ARN。有关 API 的更多信息，请参阅DescribeSolution。

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

这将显示解决方案的属性和创建 `status`。例如：

```
{
  "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. 通过自动训练，解决方案处于活动状态后，解决方案版本训练将在一小时内开始。训练开始后，您可以通过以下 ListSolutionVersions 命令获取解决方案版本的 Amazon 资源名称（ARN）：

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

4. 通过使用 `describe-solution-version` 命令来检查解决方案版本的训练 状态。提供上一步中返回的解决方案版本 ARN。有关 API 的更多信息，请参阅DescribeSolutionVersion。

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

将显示解决方案版本的属性和训练 `status`。最初，状态显示为 CREATE PENDING，例如：

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

5. 当解决方案版本 `status` 为活动时，表示训练完成。

现在，您可以使用解决方案版本，查看训练指标并创建市场活动。

> **Note**
>
> 训练需要时间。在活动中使用此版本的解决方案之前，等待训练完成（解决方案的训练状态显示为 ACTIVE）。

6. 您可通过检查解决方案版本的指标来验证解决方案的性能。通过运行以下命令来获取解决方案版本的指标。提供之前返回的解决方案版本 ARN。有关 API 的更多信息，请参阅GetSolutionMetrics。

```
aws personalize get-solution-metrics \
```

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

此时将显示示例响应：

```
{
  "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
    }
}
```

步骤 3：创建市场活动（部署解决方案）

您必须先部署解决方案版本，然后才能获得建议。部署解决方案也称为创建活动。一旦您创建活动，您的客户端应用程序就可以使用 GetRecommendations API 获得建议。

1. 通过运行以下命令来创建活动。提供上一步中返回的解决方案版本 ARN。有关 API 的更多信息，请参阅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
```

此时将显示示例响应：

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

2. 通过运行以下命令来检查部署状态。提供上一步中返回的活动 ARN。有关 API 的更多信息，请参阅DescribeCampaign。

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

此时将显示示例响应：

```
{
  "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**
>
> 在获得来自活动的推荐之前，等待 `status` 显示为“ACTIVE”。

步骤 4：获得建议

通过运行 `get-recommendations` 命令来获得建议。提供上一步中返回的活动 ARN。在请求中，指定电影评级数据集中的用户 ID。有关 API 的更多信息，请参阅GetRecommendations。

> **Note**
>
> 并非所有配方都支持 `GetRecommendations` API。有关更多信息，请参阅 选择食谱。
> 您在此步骤中调用的 AWS CLI 命令 `personalize-runtime` 与您在之前的步骤中所调用的不同。

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

在响应中，活动将返回用户可能喜欢的物品推荐（电影 ID）的列表。此列表按用户的相关性进行降序排序。

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

入门 (SDK for Python (Boto3))

本教程向您展示如何使用 SDK for Python (Boto3) 从头到尾完成 Amazon Personalize 工作流程。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

主题

- 先决条件
- 教程
- 开始使用 Jupyter（iPython）APIs 笔记本电脑的 Amazon Personalize

先决条件

以下是使用本指南中的 Python 示例的先决条件步骤：

- 完成入门先决条件，设置所需权限并创建训练数据。如果您使用自己的源数据，请确保按照先决条件设置数据格式。
- 按照中的说明设置您的 适用于 Python (Boto3) 的 AWS SDK 环境设置 AWS 软件开发工具包。

教程

在以下步骤中，您将验证您的环境并为 Amazon Personalize 创建 SDK for Python (Boto3) 客户端。然后，将导入数据，创建解决方案版本并通过市场活动进行部署，随后获得建议。

步骤 1：验证您的 Python 环境并创建 boto3 客户端

在完成先决条件之后，请运行以下 Python 示例，以确认您的环境已正确配置。此代码还会创建您在本教程中使用的 Amazon Personalize boto3 客户端。如果已正确配置环境，则会显示可用食谱列表，您现在可以运行本教程中的其他示例。

```
import boto3

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

response = personalize.list_recipes()

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

步骤 2：导入数据

创建 Amazon Personalize boto3 客户端并验证您的环境后，导入您在完成入门先决条件时创建的历史数据。要将历史数据导入 Amazon Personalize，请执行以下操作：

1. 使用以下代码创建 Amazon Personalize 架构。将 `getting-started-schema` 替换为架构的名称。

   ```
   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. 使用以下代码创建数据集组。将 dataset group name 替换为数据集组的名称。

```
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. 使用以下代码在新数据集组中创建物品交互数据集。为数据集命名，并提供前面步骤中的schema_arn和dataset_group_arn。

```
response = personalize.create_dataset(
```

```
    name = 'datase_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'Interactions'
)

dataset_arn = response['datasetArn']
```

4. 使用以下代码通过数据集导入作业导入数据。代码使用 describe_dataset_import_job 方法来跟踪作业的状态。

   将以下内容作为参数传递：作业名称、上一步中的 dataset_arn、存储训练数据的 Amazon S3 存储桶路径 (s3://*bucket name*/*folder name*/ratings.csv) 以及您的 IAM 服务角色的 ARN。您在入门先决条件期间创建了此角色。Amazon Personalize 需要权限才能访问存储桶。请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

```
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)
```

步骤 3：创建解决方案

导入数据后，创建解决方案和解决方案版本，如下所示。解决方案 包含用于训练模型的配置，解决方案版本 是经过训练的模型。

1. 使用以下代码创建新解决方案。将以下内容作为参数传递：dataset_group_arn来自之前的、解决方案的名称和 User-Personalization-v 2 配方的 ARN () arn:aws:personalize:::recipe/aws-user-personalization-v2。存储新解决方案的 ARN 以备日后使用。

```
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. 使用以下代码创建解决方案版本。将上一步中的 solution_arn 作为参数传递。以下代码将创建解决方案版本。在训练期间，代码使用 DescribeSolutionVersion 操作来检索解决方案版本的状态。训练完成后，方法将返回新解决方案版本的 ARN。请将其存储以便将来使用。

```
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)
```

步骤 4：创建市场活动

创建解决方案版本后，使用 Amazon Personalize 市场活动进行部署。使用以下代码创建可部署解决方案版本的市场活动。将以下内容作为参数传递：solution_version_arn 和市场活动名称。方法将返回新市场活动的 Amazon 资源名称 (ARN)。请将其存储以便将来使用。

```
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'])
```

步骤 5：获得建议

创建活动之后，您可以使用它来获得建议。以下代码展示了如何从市场活动中获取建议并打印出每个推荐物品的 ID。传递您在上一步中创建的市场活动的 ARN。对于用户 ID，传递来自训练数据的用户 ID，例如 123。

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

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

开始使用 Jupyter（iPython）APIs 笔记本电脑的 Amazon Personalize

要开始通过 Jupyter 笔记本使用 Amazon Personalize，请克隆或下载在 Amazon Personalize 示例存储库的 getting_started 文件夹中找到的一系列笔记本。笔记本将引导您使用 Amazon Personalize 导入训练数据、创建解决方案、创建市场活动以及获取建议。

> **Note**
>
> 在开始使用笔记本之前，请务必按照 README.md 中的步骤构建您的环境

入门 (SDK for Java 2.x)

本教程向您展示如何使用 AWS SDK for Java 2.x 从头到尾完成 Amazon Personalize 工作流程。

完成入门练习后，为避免产生不必要的费用，请删除您创建的资源。有关更多信息，请参阅 删除 Amazon Personalize 资源的要求。

有关更多示例，请参阅完成 Amazon Personalize 项目。

主题

- 先决条件
- 完成 Amazon Personalize 项目

先决条件

以下是完成本教程的先决条件步骤：

- 完成入门先决条件，设置所需权限并创建训练数据。您可以使用开始使用（控制台）或入门 (AWS CLI) 练习中所用的相同源数据。如果您使用自己的源数据，请确保按照先决条件中的步骤设置数据格式。
- 设置 SDK for Java 2.x 环境和 AWS 凭证，如《AWS SDK for Java 2.x 开发人员指南》中的设置 AWS SDK for Java 2.x 过程所述。

教程

在以下步骤中，您将项目设置为使用 Amazon Personalize 程序包并创建 Amazon Personalize SDK for Java 2.x 客户端。然后，将导入数据，创建解决方案版本并通过市场活动进行部署，随后获得建议。

步骤 1：设置您的项目以使用 Amazon Personalize 程序包

完成先决条件后，将 Amazon Personalize 依赖项添加到 pom.xml 文件中，然后导入 Amazon Personalize 程序包。

1. 将以下依赖项添加到 pom.xml 文件中。最新版本号可能与示例代码不同。

```
<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. 将以下导入语句添加至您的项目。

```
// 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;
```

步骤 2：创建 Amazon Personalize 客户端

将 Amazon Personalize 依赖项添加到 pom.xml 文件并导入所需的程序包之后，创建以下 Amazon Personalize 客户端：

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

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

步骤 3：导入数据

初始化 Amazon Personalize 客户端后，导入您在完成入门先决条件时创建的历史数据。要将历史数据导入 Amazon Personalize，请执行以下操作：

1. 将以下 Avro 架构作为 JSON 文件保存到您的工作目录中。此架构与您在完成入门先决条件时创建的 CSV 文件中的列相匹配。

```
{
  "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. 使用以下 `createSchema` 方法创建 Amazon Personalize 架构。将以下内容作为参数传递：Amazon Personalize 服务客户端、架构名称以及您在上一步中创建的架构 JSON 文件的文件路径。该方法将返回新架构的 Amazon 资源名称 (ARN)。请将其存储以便将来使用。

```
    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. 创建数据集组。使用以下 `createDatasetGroup` 方法创建数据集组。将以下内容作为参数传递：Amazon Personalize 服务客户端和数据集组的名称。该方法将返回新数据集组的 ARN。请将其存储以便将来使用。

```
    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. 创建物品交互数据集。使用以下 `createDataset` 方法创建物品交互数据集。将以下内容作为参数传递：Amazon Personalize 服务客户端、数据集名称、架构的 ARN、数据集组的 ARN 以及数据集类型的 `Interactions`。该方法将返回新数据集的 ARN。请将其存储以便将来使用。

```
    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. 使用数据集导入作业导入数据。使用以下 `createPersonalizeDatasetImportJob` 方法创建数据集导入作业。

   将以下内容作为参数传递：Amazon Personalize 服务客户端、作业名称、物品交互数据集的 ARN、存储训练数据的 Amazon S3 存储桶路径（`s3://`*`bucket name`*`/`*`folder name`*`/ratings.csv`）以及服务角色的 ARN（您在入门先决条件期间创建了此角色）。该方法将返回数据集导入作业的 ARN。（可选）将其存储起来以备后用。

```
    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 "";
```

```
    }
```

步骤 4：创建解决方案

导入数据后，创建解决方案和解决方案版本，如下所示。解决方案 包含用于训练模型的配置，解决方案版本 是经过训练的模型。

1. 使用以下 `createPersonalizeSolution` 方法创建新解决方案。将以下内容作为参数传递：Amazon Personalize 服务客户端、数据集组 Amazon 资源名称（ARN）、解决方案名称以及 User-Personalization-v2 配方的 ARN（`arn:aws:personalize:::recipe/aws-user-personalization-v2`）。该方法将返回新解决方案的 ARN。请将其存储以便将来使用。

```
   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. 使用以下 `createPersonalizeSolutionVersion` 方法创建解决方案版本。将上一步中解决方案的 ARN 作为参数传递。以下代码首先检查您的解决方案是否准备就绪，然后创建解决方案版本。在训练期间，代码使用 DescribeSolutionVersion 操作来检索解决方案版本的状态。训练完成后，方法将返回新解决方案版本的 ARN。请将其存储以便将来使用。

```
   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 "";
    }
```

有关更多信息，请参阅 手动创建解决方案版本。在您创建解决方案版本时，可以在继续前评估其性能。有关更多信息，请参阅 通过指标评估 Amazon Personalize 解决方案版本。

步骤 5：创建市场活动

训练并评估解决方案版本后，使用 Amazon Personalize 市场活动进行部署。使用以下 `createPersonalCampaign` 方法部署解决方案版本。将以下内容作为参数传递：Amazon Personalize 服务客户端、您在上一步中创建的解决方案版本的 Amazon 资源名称 (ARN) 以及市场活动名称。该方法将返回新市场活动的 ARN。请将其存储以便将来使用。

```
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);
    }
}
```

有关 Amazon Personalize 市场活动的更多信息，请参阅通过市场活动部署 Amazon Personalize 解决方案版本。

步骤 6：获得建议

创建市场活动之后，您可以使用它来获得建议。使用以下 `getRecs` 方法为用户获取建议。将以下内容作为参数传递：Amazon Personalize 运行时客户端、您在上一步中创建的市场活动的 Amazon 资源名称 (ARN) 以及您导入的历史数据中的用户 ID（例如 123）。该方法会在屏幕上打印出推荐物品列表。

```
    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);
        }
    }
```

完成 Amazon Personalize 项目

有关向您展示如何使用 SDK for Java 2.x 完成 Amazon Personalize 工作流程的多合一项目，请参阅 GitHub 上的 Amazon-Personalize-Java-App。该项目包括使用不同的食谱训练多个解决方案版本，以及使用 PutEvents 操作记录事件。

有关其他示例，请参阅在 AWS SDK 示例存储库的个性化文件夹中找到的代码。

将自己的应用场景与 Amazon Personalize 资源进行匹配

Amazon Personalize 建议可以解决以下使用案例：

- 为用户生成个性化建议
- 推荐类似或相关物品
- 推荐趋势或热门物品
- 为用户推荐下一个最佳操作（仅限使用自定义资源）
- 按相关性重新排序（仅限使用自定义资源）
- 生成用户细分（仅限使用自定义资源）

Amazon Personalize 提供基于域的资源和为这些使用案例配置的自定义资源。首先，创建域数据集组或自定义数据集组：

- 对于域数据集组，创建针对 VIDEO_ON_DEMAND 或 ECOMMERCE 域进行预配置和优化的资源。

  如果您有流视频或电子商务应用程序，我们建议您从域数据集组开始。您仍然可以添加自定义资源，例如针对自定义使用案例训练的解决方案和解决方案版本。而且，您仍可以使用自定义资源来获取批量推荐。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。
- 对于自定义数据集组，选择与您的使用案例相匹配的食谱。然后，只训练和部署可配置的解决方案与解决方案版本（经过训练的 Amazon Personalize 建议模型）。准备好之后，您可以在市场活动中部署解决方案版本来进行实时推荐。或者，您也可以在没有市场活动的情况下获取批量推荐。

  如果您没有流视频或电子商务应用程序，我们建议您创建自定义数据集组。否则，从域数据集组开始，然后根据需要添加自定义资源。

以下各部分提供了有关 Amazon Personalize 中可用的应用场景和自定义配方的详细信息。当您将自己的应用场景与 Amazon Personalize 资源进行匹配时，请注意其数据要求。在您选择应用场景或配方后，此信息有助于在为 Amazon Personalize 准备训练数据中准备数据。

主题

- 使用案例和食谱特征
- 选择使用案例
- 选择食谱

使用案例和食谱特征

对于一些使用案例和食谱，Amazon Personalize 使用以下特征来生成更相关的建议，并改善物品发现和参与度。

主题

- 实时个性化
- 探索
- 自动更新

实时个性化

对于一些使用案例和食谱，Amazon Personalize 使用实时个性化，根据用户不断变化的兴趣来更新和调整建议。当您记录用户与最近一次完整训练中出现的物品或操作交互时，它会更新针对用户的建议。您可以记录这些与事件跟踪器和 PutEvents 操作的交互，或者针对与操作的交互，可以记录与 PutActionInteractions 操作的交互。

有关记录事件的更多信息，请参阅记录实时事件以影响推荐。有关新数据影响实时建议（包括实时个性化）的信息，请参阅训练后更新数据集中的数据。

以下使用案例和食谱支持实时个性化：

- 为您推荐（ECOMMERCE 使用案例）
- 热门精选（VIDEO_ON_DEMAND 使用案例）
- User-Personalization-v2 配方
- User-Personalization 食谱
- Personalized-Ranking-v2 配方
- Personalized-Ranking 食谱
- Next-Best-Action 配方

探索

对于某些域应用场景和自定义配方，Amazon Personalize 在推荐物品时使用浏览。在浏览中，建议包括一些通常不太可能向用户建议的物品或操作，例如新物品或操作、交互量很少的物品或操作，或者根据用户以前的行为与用户不太相关的物品或操作。当您目录的更新速度非常快，或者当新物品（如新闻文章或推广）在上架之初时与用户更相关时，这可以提高物品的发现率和参与度。

配置浏览

如果您使用 User-Personalization-v2 配方，则 Amazon Personalize 会为您处理浏览配置，并且通过浏览包括的物品在推荐响应的 `Reason` 中包括 `Exploration`。为确保推荐中包括新物品，您可以根据创建时间戳使用推广筛选条件来推广新物品。有关推广的更多信息，请参阅通过实时推荐推广物品。

对于所有其他使用浏览的应用场景或配方，在创建推荐器或自定义市场活动时，或者创建批量推理作业（自定义资源）时，您可以使用以下字段配置浏览：

- 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
- 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

  要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

使用浏览的应用场景和配方

有关每个使用浏览的使用案例或食谱的更多信息，请参阅以下内容：

- 为您推荐（ECOMMERCE 使用案例）
- 热门精选（VIDEO_ON_DEMAND 使用案例）
- User-Personalization-v2 配方
- User-Personalization 食谱
- Next-Best-Action 配方

自动更新

对于某些应用场景和自定义配方，Amazon Personalize 会自动更新推荐器或解决方案版本，以考虑在推荐中加入新物品或操作。自动更新不会产生任何费用。有关包含自动更新的应用场景和配方列表，请参阅使用自动更新的域应用场景和自定义配方。

自动更新的工作方式如下：

- Amazon Personalize 何时自动更新解决方案版本或推荐器取决于您获得推荐的方式：
  - 对于实时推荐，Amazon Personalize 每两小时更新一次解决方案版本或推荐器。
  - 对于批量物品建议，当您创建批量推理作业并为解决方案指定经过全面训练的最新解决方案版本时，Amazon Personalize 会自动更新解决方案版本，以便考虑在浏览期间加入新物品。如果您不指定最新解决方案版本，则不会进行更新。
- 每次更新时，Amazon Personalize 都会使用探索，开始在推荐中包括新物品。在考虑新物品或操作时，Amazon Personalize 会考虑该物品的所有元数据。但是，只有在您记录了物品的交互并进行了完全再训练之后，这些数据才会对推荐产生更大的影响。
- 要进行更新，您必须提供自上次自动更新或再训练以来的新操作、物品或交互数据。
- 在导入 75 万个物品之前，Amazon Personalize 会考虑新物品。这是训练期间考虑的最大物品数。

自定义资源的其他指南和要求

如果您使用自定义资源，则以下是自动更新的指南和要求：

- 您的解决方案版本必须在市场活动中部署。您的市场活动会自动使用更新的解决方案版本。
- 自动更新与自动训练不同。自动更新不会创建全新的解决方案版本。而且该模型不会从您的最新数据中学习。为了维持您的解决方案，自动训练频率仍然应该是至少每周一次。
- 在解决方案自动创建新的解决方案版本，或在您手动创建新的解决方案版本之后，Amazon Personalize 不会自动更新较旧的解决方案版本，即使您已在市场活动中部署这些版本。
- 如果每两个小时的频率不够高，则您可以使用 User-Personalization 来手动创建解决方案版本，并将 `trainingMode` 设置为 `UPDATE`，以将这些新物品加入推荐中。请记住，Amazon Personalize 只会自动更新经过全面 训练的最新解决方案版本。手动更新的解决方案版本将来不会自动更新。如果您的解决方案使用自动训练，则下一个解决方案版本会恢复自动更新。如果未恢复，请手动创建一个将训练模式设置为 `FULL` 的新解决方案，然后将其部署到市场活动中。

使用自动更新的域应用场景和自定义配方

有关每个具有自动更新的使用案例或食谱的更多信息，请参阅以下内容：

- 为您推荐（ECOMMERCE 使用案例）
- 热门精选（VIDEO_ON_DEMAND 使用案例）
- User-Personalization-v2 配方

- User-Personalization 食谱
- Next-Best-Action 配方

选择使用案例

在域数据集组中创建推荐器时，您需指定使用案例。Amazon Personalize 通过使用案例的最佳配置，训练支持推荐器的模型。每个域都有不同的使用案例。例如，如果您为域数据集组指定 VIDEO_ON_DEMAND，则只有 VIDEO_ON_DEMAND 使用案例可用。每个使用案例对获取建议都有不同的要求。某些使用案例需要特定的事件类型。您可以自由添加其他事件类型。

对于所有使用案例，您的交互数据必须包含以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个唯一的用户 ID，每位用户至少有两次物品交互。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

主题

- VIDEO_ON_DEMAND 使用案例
- ECOMMERCE 使用案例

VIDEO_ON_DEMAND 使用案例

以下部分列出了每个 VIDEO_ON_DEMAND 使用案例的要求和 Amazon 资源名称 (ARN)。对于所有使用案例，您的交互数据必须包含以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个独立用户 IDs ，每个用户至少有两次物品互动。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

> **Note**
>
> 如果您使用 CreateRecommender API，请提供此处列出的食谱 ARN。

主题

- 因为您观看过 X
- 更喜欢 X
- 最受欢迎
- 当前趋势
- 热门精选

因为您观看过 X

根据您指定的视频，获取其他用户也观看过的视频建议。对于此使用案例，Amazon Personalize 会根据您指定的 userId 和 `Watch` 事件自动筛选用户观看过的视频。如果您应用自己的筛选器，则筛选器将在用户观看过的视频被筛选掉之后应用。

筛选时，Amazon Personalize 会考虑每位用户每种事件类型最多 100 次物品交互。这适用于任何自动或自定义筛选器。您可以使用服务限额控制台请求增加该限额。有关更多信息，请参阅《服务限额用户指南》中的请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-vod-because-you-watched-x`
- GetRecommendations API 要求：

  `userId`：必需

  `itemId`：必需
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 `Watch` 事件。

更喜欢 X

获取与您指定的视频相似的视频的建议。对于此使用案例，Amazon Personalize 会根据您指定的 userId 和 Watch 事件自动筛选用户购买过的视频。如果您应用自己的筛选器，则筛选器将在用户观看过的视频被筛选掉之后应用。

筛选时，Amazon Personalize 会考虑每位用户每种事件类型最多 100 次物品交互。这适用于任何自动或自定义筛选器。您可以使用服务限额控制台请求增加该限额。有关更多信息，请参阅《服务限额用户指南》中的请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

- 食谱 ARN：arn:aws:personalize:::recipe/aws-vod-more-like-x
- GetRecommendations API 要求：

  userId：必需

  itemId：必需
- 训练时使用的数据集：
  - 交互（必需）
  - 物品（必需）
- 所需事件数量：至少 1000 个任意类型的事件。
- 建议的事件类型：Watch 和 Click 事件。

最受欢迎

获取有关用户观看次数最多的视频的建议。

- 食谱 ARN：arn:aws:personalize:::recipe/aws-vod-most-popular
- GetRecommendations要求：

  userId：必需

  itemId：未使用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 Watch 事件。

当前趋势

获取有关当前热门视频的建议。热门视频是指在用户中迅速变得越来越受欢迎的内容。每隔两小时，Amazon Personalize 会自动评估一次您的交互数据，并识别热门物品。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-vod-trending-now`
- GetRecommendations API 要求：

  `userId`: 仅当您按 CurrentUser 或按用户互动过的项目进行筛选时才为必填项

  `itemId`：未使用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 所需事件数量：至少 1000 个任意类型的事件。

热门精选

为您指定的用户获取个性化内容建议。对于此使用案例，Amazon Personalize 会根据您指定的 userId 和 `Watch` 事件自动筛选用户购买过的视频。如果您应用自己的筛选器，则筛选器将在用户观看过的视频被筛选掉之后应用。

筛选时，Amazon Personalize 会考虑每位用户每种事件类型最多 100 次物品交互。这适用于任何自动或自定义筛选器。您可以使用服务限额控制台请求增加该限额。有关更多信息，请参阅《服务限额用户指南》中的请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

在推荐项目时，此用例会使用real-time-personalization和探索。它还使用自动更新来考虑推荐新物品。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-vod-top-picks`
- GetRecommendations 要求：

  `userId`：必需

  `itemId`：未使用
- 训练时使用的数据集：
  - 交互（必需）
  - 物品（可选）
  - 用户（可选）
- 所需事件数量：至少 1000 个事件。

- 建议的事件类型：`Click` 和 `Watch` 事件。
- 浏览配置参数：创建推荐器时，可以使用以下内容配置浏览。
  - 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
  - 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

    要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

ECOMMERCE 使用案例

以下部分列出了每个 ECOMMERCE 使用案例的要求和 Amazon 资源名称 (ARN)。对于所有使用案例，您的交互数据必须包含以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个独立用户 IDs ，每个用户至少有两次物品互动。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

> **Note**
>
> 如果您使用 CreateRecommender API，请提供此处列出的食谱 ARN。

主题

- 最常查看
- 最畅销
- 经常一起购买

- 浏览 X 的客户也浏览过
- 为您推荐

最常查看

根据客户查看某件物品的次数获取热门物品建议。

- 食谱 ARN：arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-views
- GetRecommendations 要求：

  userId：必需

  itemId：未使用

  inputList：不适用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 View 事件。

最畅销

根据客户购买某件物品的次数获取热门物品建议。

- 食谱 ARN：arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-purchases
- GetRecommendations 要求：

  userId：必需

  itemId：未使用

  inputList：不适用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 Purchase 事件。

经常一起购买

获取客户经常一起购买的物品以及您指定的物品的建议。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-ecomm-frequently-bought-together`
- GetRecommendations 要求：

  `userId`: 仅当你按以下条件筛选时才需要填写 CurrentUser

  `itemId`：必需

  `inputList`：不适用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 `Purchase` 事件。

浏览 X 的客户也浏览过

根据您指定的物品，获取客户也查看过的物品建议。对于此使用案例，Amazon Personalize 会根据您指定的用户 ID 和 `Purchase` 事件自动筛选用户购买的物品。如果您应用自己的筛选器，则筛选器将在用户已购买的物品被筛选掉之后应用。

筛选时，Amazon Personalize 会考虑每位用户每种事件类型最多 100 次物品交互。这适用于任何自动或自定义筛选器。您可以使用服务限额控制台请求增加该限额。有关更多信息，请参阅《服务限额用户指南》中的请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-ecomm-customers-who-viewed-x-also-viewed`
- GetRecommendations 要求：

  `userId`：必需

  `itemId`：必需

  `inputList`：不适用
- 训练时使用的数据集：仅限物品交互数据集（必需）
- 必需的事件类型：至少 1000 个 `View` 事件。
- 建议的事件类型：`Purchase` 事件。

为您推荐

根据您指定的用户获取针对物品的个性化建议。对于此使用案例，Amazon Personalize 会根据您指定的用户 ID 和 Purchase 事件自动筛选掉用户购买的物品。如果您应用自己的筛选器，则筛选器将在用户已购买的物品被筛选掉之后应用。

筛选时，Amazon Personalize 会考虑每位用户每种事件类型最多 100 次物品交互。这适用于任何自动或自定义筛选器。您可以使用服务限额控制台请求增加该限额。有关更多信息，请参阅《服务限额用户指南》中的请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

在推荐项目时，此用例会使用real-time-personalization和探索。它还使用自动更新来考虑推荐新物品。

- 食谱 ARN：`arn:aws:personalize:::recipe/aws-ecomm-recommended-for-you`
- GetRecommendations 要求：

  `userId`：必需

  `itemId`：未使用

  `inputList`：不适用
- 训练时使用的数据集：
  - 交互（必需）
  - 物品（可选）
  - 用户（可选）
- 所需事件数量：至少 1000 个事件。
- 建议的事件类型：`View` 和 `Purchase` 事件。
- 浏览配置参数：创建推荐器时，可以使用以下内容配置浏览。
  - 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
  - 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

选择食谱

当您创建自定义解决方案时，您需指定食谱并配置训练参数。食谱 是针对特定使用案例准备的 Amazon Personalize 算法。Amazon Personalize 根据常见使用案例为训练模型提供食谱。当您为解决方案创建解决方案版本时，Amazon Personalize 会根据食谱和训练配置训练支持该解决方案版本的模型。

Amazon Personalize 食谱在训练期间使用以下内容：

- 数据的预定义属性
- 预定义特征转换
- 预定义算法
- 算法的初始参数设置

为了优化模型，您可以在创建解决方案时覆盖上述多个参数。有关更多信息，请参阅 超级参数和 HPO。

主题

- 按使用案例划分的 Amazon Personalize 食谱类型
- Amazon Personalize 食谱
- 查看可用的 Amazon Personalize 食谱
- User-Personalization-v2 配方
- User-Personalization 食谱
- Trending-Now 食谱
- Popularity-Count 食谱
- Personalized-Ranking-v2 配方
- Personalized-Ranking 食谱
- 语义相似度配方
- Similar-Items 食谱

- SIMS 配方
- Next-Best-Action 配方
- Item-Affinity 食谱
- Item-Attribute-Affinity 食谱
- 旧版 HRNN 配方

按使用案例划分的 Amazon Personalize 食谱类型

要选择您的食谱，请先从以下内容中选择您的使用案例，并记下其对应的食谱类型。

- 向用户推荐物品（USER_PERSONALIZATION 食谱）

  要向用户提供个性化建议，请使用 USER_PERSONALIZATION 食谱训练您的模型。个性化建议有助于提高参与度和转化率。

- 对用户物品排名（PERSONALIZED_RANKING 食谱）

  要对用户的精选列表或搜索结果顺序进行个性化处理，请使用 PERSONALIZED_RANKING 食谱训练您的模型。PERSONALIZED_RANKING 食谱根据给定用户的预测兴趣水平，对一组输入项进行重新排名，从而创建个性化列表。个性化列表可改善客户体验，提高客户忠诚度和参与度。

- 推荐趋势或热门物品（POPULAR_ITEMS 食谱）

  要推荐趋势或热门物品，请使用 POPULAR_ITEMS 食谱。如果您的客户高度重视与其他用户交互的内容，则您可以使用 POPULAR_ITEMS。常见用途包括推荐病毒式社交媒体内容、突发新闻文章或最近的体育视频。

- 推荐类似物品（RELATED_ITEMS 食谱）

  要推荐类似物品（例如，经常一起购买的物品或其他用户也看过的电影），您应该使用 RELATED_ITEMS 食谱。推荐类似物品可以帮助您的客户发现物品，还可以提高用户转化率。

- 建议下一个最佳操作（PERSONALIZED_ACTIONS 配方）

  要实时向用户建议下一个最佳操作，例如注册忠诚度计划或申请信用卡，您应该使用 PERSONALIZED_ACTIONS 配方。建议下一个最佳操作可以提高客户忠诚度、创造更多收入并改善用户体验。

- 获取用户细分（USER_SEGMENTATION 食谱）

要根据物品输入数据获取用户细分（例如，最有可能与具有特定属性的物品进行交互的用户），您应该使用 USER_SEGMENTATION 食谱。获取用户细分可以帮助您创建高级市场活动，根据不同的用户细分采取行动的可能性，将不同的物品推广给这类用户细分。

Amazon Personalize 食谱

Amazon Personalize 提供以下类型的食谱。除了行为差异外，每种类型对获取推荐都有不同的要求，如下表所示。

食谱类型	配方	API	API 要求
USER_PERSONALIZATION	User-Personalization-v2 User-Personalization HRNN 食谱（旧版） HRNN-Metadata 食谱（旧版） HRNN-Coldstart 食谱（旧版）	GetRecommendations	`userId`：必需 `itemId`：未使用 `inputList`：不适用
POPULAR_ITEMS	Trending-Now 热门程度-计数	GetRecommendations	`userId`：仅在您应用了需要它的过滤器时才是必需的 `itemId`：未使用 `inputList`：不适用
PERSONALIZED_RANKING	Personalized-Ranking-v2 Personalized-Ranking	GetPersonalizedRanking	`userId`：必需 `itemId`：不适用

食谱类型	配方	API	API 要求
			`inputList` ：itemId 的列表
RELATED_ITEMS	Similar-Items SIMS 语义相似度配方	GetRecommendations	`userId`：仅在您应用了需要它的过滤器时才是必需的 `itemId`：必需 `inputList` ：不适用
PERSONALIZED_ACTIONS	Next-Best-Action	GetActionRecommendations	`userId`：必需 `actionId`：未使用 `itemId`：未使用 `inputList` ：不适用
USER_SEGMENTATION	Item-Affinity Item-Attribute-Affinity	CreateBatchSegmentJob	有关批量工作流程的要求，请参阅使用批量细分作业来获取用户细分。

查看可用的 Amazon Personalize 食谱

查看可用配方列表：

- 在 Amazon Personalize 控制台中，选择一个数据集组。在导航窗格中，选择解决方案和配方，然后选择配方选项卡。
- 使用后 适用于 Python (Boto3) 的 AWS SDK，调用 ListRecipes API。
- 对于 AWS CLI，使用以下命令。

```
aws personalize list-recipes
```

要使用 SDK for Python (Boto3) 获取有关食谱的信息，请调用 DescribeRecipe API。要使用获取有关配方的信息 AWS CLI，请使用以下命令。

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

User-Personalization-v2 配方

User-Personalization-v2（aws-user-personalization-v2）配方会根据用户的偏好推荐他们将与之交互的物品。例如，您可以使用 User-Personalization-v2 为流式传输应用程序生成个性化影片推荐，或者为零售应用程序生成个性化产品推荐。其他应用场景包括为新闻网站生成实时推荐或为个性化营销活动生成批量推荐。

User-Personalization-v2 可以对物品交互数据集和物品数据集中的多达 500 万个物品进行训练。而且，它以比 User-Personalization 更低的延迟生成更相关的推荐。

因为 User-Personalization-v2 会根据您的数据向用户推荐最相关的物品，所以它会更频繁地推荐带有交互数据的现有物品。为确保推荐中包括新物品，您可以根据创建时间戳使用包括一些物品的推广。有关推广的更多信息，请参阅通过实时推荐推广物品。

此配方使用基于转换器的架构来训练一个模型，该模型可以了解上下文并跟踪数据中的关系和模式。转换器是一种神经网络架构，可将输入序列转换或更改为输出序列。对于 Amazon Personalize，输入序列是用户在您的数据中的物品交互历史记录。输出序列是他们的个性化推荐。有关转换器的更多信息，请参阅 AWS 云计算概念中心的什么是人工智能中的转换器？。

User-Personalization-v2 使用与其他配方不同的定价模式。有关定价的更多信息，请参阅 Amazon Personalize 定价。

主题

- 配方特征
- 必需和可选数据集

- 属性和超参数

配方特征

User-Personalization-v2 在生成物品推荐时使用以下 Amazon Personalize 配方特征：

- 实时个性化 - 使用实时个性化时，Amazon Personalize 会根据用户不断变化的兴趣来更新和调整物品推荐。有关更多信息，请参阅 实时个性化。
- 浏览 - 使用浏览时，推荐包括的物品具有更少的交互数据或用户相关性。使用 User-Personalization-v2 时，Amazon Personalize 会为您处理浏览配置。为确保推荐中包括新物品，您可以根据其创建时间戳使用包括新物品的推广。有关推广的更多信息，请参阅通过实时推荐推广物品。
- 自动更新 - 通过自动更新，Amazon Personalize 每两小时自动更新一次最新模型（解决方案版本），以便为推荐考虑新物品。有关更多信息，请参阅 自动更新。
- 推荐元数据 - 使用 User-Personalization-v2 配方时，如果您的物品数据集包含至少一列元数据，则市场活动可以自动选择在推荐结果中包括物品元数据。您没有手动为市场活动启用元数据。您可以使用元数据来丰富用户界面中的建议，例如将电影的类型添加到轮播中。有关更多信息，请参阅 推荐中的物品元数据。

必需和可选数据集

要使用 User-Personalization-v2，您必须创建一个物品交互数据集，并导入至少 1000 个物品交互。Amazon Personalize 主要根据物品交互数据生成推荐。有关更多信息，请参阅 物品交互数据。User-Personalization-v2 可以对物品交互数据集和物品数据集中的多达 500 万个物品进行训练。

使用 User-Personalization-v2 时，Amazon Personalize 可以使用包含以下内容的物品交互数据：

- 事件类型和事件值数据 - Amazon Personalize 使用事件类型数据（例如点击或观看事件类型），通过用户行为中的任何模式来识别用户的意图和兴趣。此外，您还可以在训练前使用事件类型和事件值数据来筛选记录。有关更多信息，请参阅 事件类型和事件值数据。

  > **Note**
  >
  > 使用 User-Personalization-v2 时，在按事件类型或值进行筛选之前，训练费用将基于交互数据。有关定价的更多信息，请参阅 Amazon Personalize 定价。

- 上下文元数据 - 上下文元数据是您在事件发生时在用户环境中收集的交互数据，例如其位置或设备类型。有关更多信息，请参阅 上下文元数据。

以下数据集是可选的，可以用于改进推荐：

- 用户数据集 - Amazon Personalize 可以使用您的用户数据集中的数据来更好地了解您的用户及其兴趣。您还可以使用用户数据集中的数据来筛选推荐。有关您可以导入的用户数据的信息，请参阅用户元数据。
- 物品数据集 - Amazon Personalize 可以使用您的物品数据集中的数据来识别其行为中的关联和模式。这有助于 Amazon Personalize 了解您的用户及其兴趣。您还可以使用物品数据集中的数据来筛选推荐。有关您可以导入的物品数据的信息，请参阅物品元数据。

属性和超参数

User-Personalization-v2 配方有以下属性：

- 名称 – `aws-user-personalization-v2`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-user-personalization-v2`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-user-personalization-v2`

有关更多信息，请参阅 选择食谱。

下表描述了 User-Personalization-v2 配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。为超参数选择最佳值的过程称为超参数优化 (HPO)。使用 User-Personalization-v2 时，如果开启自动训练，则 Amazon Personalize 将每 90 天自动执行一次 HPO。如果没有开启自动训练，就不会执行 HPO。

该表为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）

名称	描述
算法超参数	
`apply_recency_bias`	确定模型是否应为物品交互数据集中的最新物品交互数据赋予更多权重。最新的交互数据可能包括交互事件底层模式的突然变化。

名称	描述
	要训练一个使最近事件具有更高权重的模型，请将 apply_recency_bias 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 apply_recency_bias 设置为 false。 默认值：true 范围：true 或 false 值类型：布尔值 HPO 可调：否

User-Personalization 食谱

> **Important**
>
> 我们建议使用 User-Personalization-v2 配方。它可以通过更快的训练考虑多达 500 万个物品，并以更低的延迟生成相关的推荐。

用户个性化 (aws-user-personalization) 配方针对所有个性化推荐场景进行了优化。它可预测用户最有可能与之交互的物品。您可以使用 User-Personalization 为流式传输应用程序生成个性化影片推荐，或者为零售应用程序生成个性化产品推荐。

使用 User-Personalization 时，Amazon Personalize 主要根据物品交互数据集中的用户物品交互数据来生成推荐。它还可以使用物品数据集和用户数据集中的任何物品和用户元数据。有关它使用的数据的更多信息，请参阅必需和可选数据集。

主题

- 配方特征
- 必需和可选数据集
- 属性和超参数
- 使用 User-Personalization 食谱进行训练（控制台）
- 使用 User-Personalization 食谱进行训练 (Python SDK)

- 获取建议并记录展示次数 (SDK for Python (Boto3))
- 示例 Jupyter 笔记本

配方特征

User-Personalization 在生成物品推荐时使用以下 Amazon Personalize 配方特征：

- 实时个性化 - 使用实时个性化时，Amazon Personalize 会根据用户不断变化的兴趣来更新和调整物品推荐。有关更多信息，请参阅 实时个性化。
- 浏览 - 通过浏览，推荐包括新物品或交互数据较少的物品。当您目录的更新速度非常快，或者当新物品（如新闻文章或推广）在上架之初时与用户更相关时，这可以提高物品的发现率和参与度。有关浏览的更多信息，请参阅探索。
- 自动更新 - 通过自动更新，Amazon Personalize 每两小时自动更新一次最新模型（解决方案版本），以便为推荐考虑新物品。有关更多信息，请参阅 自动更新。

必需和可选数据集

要使用 User-Personalization，您必须创建一个物品交互数据集，并导入至少 1000 个物品交互。Amazon Personalize 主要根据物品交互数据生成推荐。

使用 User-Personalization 时，Amazon Personalize 可以使用包含以下内容的物品交互数据：

- 事件类型和事件值数据 - Amazon Personalize 使用事件类型数据（例如点击或观看事件类型），通过用户行为中的任何模式来识别用户的意图和兴趣。此外，您还可以在训练前使用事件类型和事件值数据来筛选记录。有关更多信息，请参阅 事件类型和事件值数据。
- 上下文元数据 - 上下文元数据是您在事件发生时在用户环境中收集的交互数据，例如其位置或设备类型。有关更多信息，请参阅 上下文元数据。
- 展示数据 - 展示是用户与特定物品交互（点击、观看、购买等）时可见的物品列表。有关更多信息，请参阅 展示数据。

以下数据集是可选的，可以用于改进推荐：

- 用户数据集 - Amazon Personalize 可以使用您的用户数据集中的数据来更好地了解您的用户及其兴趣。您还可以使用用户数据集中的数据来筛选推荐。有关您可以导入的用户数据的信息，请参阅用户元数据。

- 物品数据集 - Amazon Personalize 可以使用您的物品数据集中的数据来识别其行为中的关联和模式。这有助于 Amazon Personalize 了解您的用户及其兴趣。您还可以使用物品数据集中的数据来筛选推荐。有关您可以导入的物品数据的信息，请参阅物品元数据。

属性和超参数

User-Personalization 食谱具有以下属性：

- 名称 – `aws-user-personalization`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-user-personalization`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-user-personalization`

有关更多信息，请参阅 选择食谱。

下表描述了 User-Personalization 食谱的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO？

名称	说明
算法超参数	
`hidden_dimension`	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当物品交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。要确定最佳值，请使用 HPO。要使用 HPO，当您调用 CreateSolution 和 CreateSolutionVersion 操作时，请将 `performHPO` 设置为 `true`。

名称	说明
	默认值：149 范围：[32, 256] 值类型：整数 HPO 可调：是
bptt	确定是否使用基于时间的反向传播技术。反向传播 是一种在基于递归神经网络的算法中更新权重的技术。将 bptt 用于长期积分以将延迟奖励与早期事件联系起来。例如，延迟奖励可以是几次点击后进行的购买。早期事件可以是初始点击。即使在相同的事件类型（例如点击）中，最好考虑长期影响并最大化总奖励。要考虑长期效果，请使用较大的 bptt 值。使用较大的 bptt 值需要更大的数据集和更多的处理时间。 默认值：32 范围：[2, 32] 值类型：整数 HPO 可调：是

名称	说明
recency_mask	确定模型是否应考虑物品交互数据集中的最新流行趋势。最新流行趋势可能包括交互事件的基本模式的突然变化。要训练一个使最近事件具有更高权重的模型，请将 recency_mask 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 recency_mask 设置为 false。要使用相等权重获得好的建议，您可能需要较大的训练数据集。 默认值：True 范围：True 或 False 值类型：布尔值 HPO 可调：是
特征化超参数	

名称	说明
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是有关用户的数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min_user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	说明
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是有关用户的数据总量。使用 max_user_history_length_percentile 排除具有较长历史记录长度的一定百分比的用户，因为这些用户的数据往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min_user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.99 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否
物品浏览市场活动配置超参数	
exploration_weight	确定建议包含物品交互数据较少或相关性较低的物品的频率。值越接近 1.0，浏览次数就越多。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。有关更多信息，请参阅 the section called “Campaign Config”。 默认值：0.3 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	说明
exploration_item_age_cut_of f	指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。 要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。 默认值：30.0 范围：正浮点数 值类型：浮点数 HPO 可调：否

使用 User-Personalization 食谱进行训练（控制台）

要使用 User-Personalization 食谱在控制台中生成建议，请先使用食谱训练新的解决方案版本。然后，使用解决方案版本部署市场活动并使用市场活动获取建议。

使用 User-Personalization 食谱训练新解决方案版本（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 使用新架构创建自定义数据集组，然后上传包含展示数据的数据集。（可选）在物品数据集中包含 CREATION_TIMESTAMP 和非结构化文本元数据数据，这样，Amazon Personalize 就可以更准确地计算物品的使用期限并识别冷物品。

   有关导入数据的更多信息，请参阅将训练数据导入 Amazon Personalize 数据集。

3. 在数据集组页面上，选择包含一个或多个带有展示数据的数据集的新数据集组。
4. 在导航窗格中，选择解决方案和食谱，然后选择创建解决方案。
5. 在创建解决方案页面上，对于解决方案名称，输入新解决方案的名称。
6. 对于解决方案类型，选择物品建议，为用户获取物品建议。
7. 在配方中，选择 aws-user-personalization。这时，将显示解决方案配置部分，其中提供了几个配置选项。
8. 在事件配置中，如果您的项目交互数据集包含 EVENT_TYPE 或同时包含 EVENT_TYPE 和 EVENT_VALUE 列，则可以选择使用事件类型和事件值阈值字段，来选择 Amazon Personalize 在训练模型时使用的项目交互数据。有关更多信息，请参阅 选择用于训练的物品交互数据。

   如果您有多个事件类型并使用 User-Personalization-v 2 个配方或 Personalized-Ranking-v 2 个配方，则也可以为不同的类型指定不同的权重。例如，您可以配置一个解决方案，为购买事件赋予高于点击事件的权重。有关更多信息，请参阅 使用事件配置优化解决方案。
9. （可选）为您的解决方案配置超参数。有关 User-Personalization 食谱属性和超参数的列表，请参阅属性和超参数。
10. 选择创建并训练解决方案以开始训练。此时，将显示控制面板页面。

    您可以导航到解决方案详情页面，在解决方案版本 部分跟踪训练进度。训练完成时，状态会变为活动。

创建市场活动并获取建议（控制台）

当您的解决方案版本状态为活动时，您就可以按如下方式创建市场活动并获得建议了：

1. 在解决方案详细信息页面或市场活动 页面上，选择创建新的市场活动。
2. 在创建新的市场活动页面上，对于市场活动详细信息，提供以下信息：
   - 市场活动名称：输入市场活动的名称。您在此处输入的文本将显示在市场活动控制面板和详细信息页面上。
   - 解决方案：选择您刚刚创建的解决方案。
   - 解决方案版本 ID：选择您刚刚创建的解决方案版本 ID。
   - 最低预调配每秒事务数：设置 Amazon Personalize 支持的最低预调配每秒事务数。有关更多信息，请参阅 CreateCampaign 操作。
3. 对于活动配置，提供以下信息：

- 浏览权重：配置要浏览的程度，其中，建议包含的物品交互数据越少或相关性越低，您所指定的浏览频率就越高。值越接近 1，浏览次数就越多。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
- 浏览物品期限截止：输入自上次交互以来的最大物品期限（以天为单位），以定义物品浏览的范围。要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。

  例如，如果您输入 10，则在浏览期间仅考虑数据集中自最近一次交互以来 10 天内具有物品交互数据的物品。

  > **Note**
  >
  > 建议可能包括没有来自该时间范围之外的物品交互数据的物品。这是因为这些物品与用户兴趣相关，但不需要浏览识别它们。

4. 选择创建市场活动。
5. 在市场活动详细信息页面上，当市场活动状态变为活动 时，您可以使用市场活动来获取建议并记录展示量。有关更多信息，请参阅“入门”中的步骤 5：获得建议。

   Amazon Personalize 会每两小时自动更新一次最新解决方案版本，以包含新数据。您的市场活动会自动使用更新的解决方案版本。有关更多信息，请参阅 自动更新。

   要手动更新市场活动，请先使用控制台或 CreateSolutionVersion 操作创建和训练新的解决方案版本，并将 `trainingMode` 设置为 `update`。然后，在控制台的市场活动 页面或使用 UpdateCampaign 操作，手动更新市场活动。

   > **Note**
   >
   > Amazon Personalize 不会自动更新您在 2020 年 11 月 17 日之前创建的解决方案版本。

使用 User-Personalization 食谱进行训练 (Python SDK)

当您创建了数据集组并上传了包含展示数据的数据集后，您可以使用 User-Personalization 食谱训练解决方案。（可选）在物品数据集中包含 CREATION_TIMESTAMP 和非结构化文本元数据数据，这样，Amazon Personalize 就可以更准确地计算物品的使用期限并识别冷物品。有关创建数据集组和上传训练数据的更多信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。

使用 SDK 使用用户个性化配方训练解决方案 AWS

1. 使用 create_solution 方法创建新的解决方案。

   将 solution name 替换为您的解决方案名称，将 dataset group arn 替换为数据集组的 Amazon 资源名称 (ARN)。

```
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)
```

   有关 aws-user-personalization配方属性和超参数的列表，请参见属性和超参数。

2. 使用更新的训练数据创建新的解决方案版本，并使用以下代码片段将 trainingMode 设置为 FULL。将 solution arn 替换为您的解决方案 ARN。

```
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. 当 Amazon Personalize 创建完解决方案版本后，使用以下参数创建市场活动：

   - 提供新的 campaign name 及步骤 2 中生成的 solution version arn。
   - 修改 explorationWeight 物品浏览配置超参数以配置要浏览的程度。值越接近 1.0，建议物品交互数据较少或相关性较低的物品的频率就越高。默认值为 0.3。

- 修改 explorationItemAgeCutOff 物品浏览配置超参数，以提供应浏览物品的最大持续时间（相对于最新交互的天数）。该值越大，浏览期间考虑的物品就越多。

使用以下 Python 片段创建新市场活动，其以浏览为重点，浏览截止日期为 30 天。创建市场活动通常需要几分钟，但也可能需要一个多小时。

```
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)
```

通过 User-Personalization，Amazon Personalize 每两小时自动更新一次解决方案版本，以包含新数据。您的市场活动会自动使用更新的解决方案版本。有关更多信息，请参阅 自动更新。

要手动更新市场活动，请先使用控制台或 CreateSolutionVersion 操作创建和训练新的解决方案版本，并将 trainingMode 设置为 update。然后，在控制台的市场活动 页面或使用 UpdateCampaign 操作，手动更新市场活动。

> **Note**
>
> Amazon Personalize 不会自动更新您在 2020 年 11 月 17 日之前创建的解决方案版本。

获取建议并记录展示次数 (SDK for Python (Boto3))

创建市场活动后，您可以使用它为用户获取建议并记录展示次数。有关使用获取批次推荐的信息，AWS SDKs 请参阅创建批量推理作业 (AWS SDKs)。

获取建议并记录展示次数

1. 调用 get_recommendations 方法。将 campaign arn 更改为新市场活动的 ARN，将 user id 更改为用户的 userId。

```
import boto3

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

2. 创建用于发送 PutEvents 请求的新事件跟踪器。将 event tracker name 替换为事件跟踪器的名称，将 dataset group arn 替换为数据集组的 ARN。

```
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. 使用步骤 1 中的 recommendationId 和步骤 2 中的 event tracking id 来创建新 PutEvents 请求。此请求记录来自用户会话的新展示数据。将 user id 更改为用户的 ID。

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

示例 Jupyter 笔记本

有关展示如何使用 User-Personalization 食谱的示例 Jupyter 笔记本，请参阅具有浏览的 User Personalization。

Trending-Now 食谱

Trending-Now 配方 (aws-trending-now) 会针对在用户中迅速变得越来越受欢迎的商品生成推荐。如果越来越受欢迎的物品与您的客户更相关，则您可以使用 Trending-Now 食谱。例如，您的客户可能高度重视其他用户交互的内容。常见用途包括推荐病毒式社交媒体内容、突发新闻文章或最近的体育视频。

Trending-Now 通过计算每个物品在可配置的时间间隔内的交互增加量来自动识别热门物品。增长率最高的物品被视为热门物品。时间基于物品交互数据集中的时间戳数据。考虑的物品来自您批量和增量导入的交互数据。您无需为 Trending-Now 手动创建新的解决方案版本，即可考虑交互数据中的新物品。

您可以在创建解决方案时提供 `Trend discovery frequency`，以指定时间间隔。例如，如果您为 `Trend discovery frequency` 指定 `30 minutes`，则对于每 30 分钟的数据，Amazon Personalize 会识别自上次评估以来交互增长率最高的物品。可能的频率包括 30 分钟、1 小时、3 小时和 1 天。选择与您的交互数据分布一致的频率。在您选择的时间间隔内缺少数据会降低建议的准确性。如果您在过去两个时间间隔内导入的交互数据为零，则 Amazon Personalize 将只推荐受欢迎的物品，而不推荐热门物品。

使用 Trending-Now，您可以在 Amazon Personalize 控制台的测试市场活动页面上，调用 GetRecommendations 操作或获取建议。Amazon Personalize 会返回最热门的物品。只有在您应用需要 `userId` 的筛选器时，才需在请求中传递该 ID。使用 GetRecommendations API，您可以使用`numResults`参数配置返回的热门项目数量。您无法通过 Trending-Now 食谱获得批量建议。

要使用 Trending-Now，您必须创建一个项目交互数据集，其中包含至少 1000 个唯一的历史和事件互动（按事件类型筛选后，如果提供）。 eventValueThreshold在生成热门物品推荐时，Trending-Now 不使用物品或用户数据集中的数据。但是，您仍然可以根据这些数据集中的数据筛选建议。有关更多信息，请参阅 筛选建议和用户细分。

主题

- 属性和超参数

- 创建解决方案 (SDK for Python (Boto3))
- 示例 Jupyter 笔记本

属性和超参数

Trending-Now 食谱具有以下属性：

- 名称 – `aws-trending-now`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-trending-now`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-trending-now-custom`

有关更多信息，请参阅 选择食谱。

下表描述 Trending-Now 食谱的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO？

名称	说明
特征转换超参数	
`Trend discovery frequency`	指定 Amazon Personalize 评估交互数据并识别热门物品的频率。例如，如果您为 `Trend discovery frequency` 指定 `30 minutes`，则每隔 30 分钟，Amazon Personalize 就会识别 30 分钟时间间隔内交互增长率最高的物品。 可用的频率包括 30 分钟、1 小时、2 小时、3 小时和 1 天。选择与您的交互数据分布一致的频率。在您选择的时间间隔内缺少数据会降低建议的准确性。如果您使用

名称	说明
	CreateSolution API 操作但未指定值，则默认值为每 2 小时一次。 **Note** 对于所有大于 2 小时的参数值，Amazon Personalize 会每 2 小时自动刷新一次热门项目推荐，以涵盖新交互和新项目。 默认值：2 小时 可能的值：30 分钟、1 小时、2 小时、3 小时和 1 天。 值类型：字符串 HPO 可调：否

创建解决方案 (SDK for Python (Boto3))

以下代码显示了如何使用 SDK for Python (Boto3) 创建具有 Trending-Now 食谱的解决方案。`trend_discovery_frequency` 的可能值为 `30 minutes`、`1 hour`、`3 hours` 和 `1 day`。有关使用控制台创建解决方案的信息，请参阅创建解决方案（控制台）。

```
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'])
```

示例 Jupyter 笔记本

有关展示如何使用 Trending-Now 食谱的 Jupyter 笔记本示例，请参阅 Amazon Personalize 示例存储库中的 trending _now_example. ipynb。 GitHub

Popularity-Count 食谱

Popularity-Count 根据您的交互数据推荐最受欢迎的物品。最受欢迎的物品是唯一用户交互数据最多的物品。配方为所有用户返回相同的热门项目。Popularity-Count 是一个不错的基准，用于使用 Amazon Personalize 在您创建解决方案版本时生成的评估指标与其他食谱进行比较。有关更多信息，请参阅 通过指标评估 Amazon Personalize 解决方案版本。

创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 Popularity-Count 时，您必须手动创建新的解决方案版本（重新训练模型），以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。然后，您必须使用解决方案版本更新所有市场活动。有关更多信息，请参阅 维护建议相关性。

此预定义配方具有以下属性：

- 名称 – `aws-popularity-count`
- 食谱 ARN - `arn:aws:personalize:::recipe/aws-popularity-count`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-popularity-count`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/sims`
- 食谱类型 - `USER_PERSONALIZATION`

Popularity-Count 没有公开的超参数。

Personalized-Ranking-v2 配方

Personalized-Ranking-v2 配方可生成物品的个性化排名。个性化排名 是为特定用户按相关性重新排名的推荐物品的列表。如果您有一系列已订购的物品（例如搜索结果、推广或精选列表），且想要为每位用户提供个性化重新排名，则此食谱非常有用。

Personalized-Ranking-v2 可以对物品交互数据集和物品数据集中的多达 500 万个物品进行训练。而且它以比Personalized-Ranking更低的延迟生成更准确的排名。

使用 Personalized-Ranking-v2 时，您可以在 GetPersonalizedRanking API 操作中指定要排名的物品。如果您指定的物品没有交互数据，则 Amazon Personalize 将在 GetPersonalizedRanking API 响应中返回这些没有推荐分数的物品。

此配方使用基于转换器的架构来训练一个模型，该模型可以了解上下文并跟踪数据中的关系和模式。转换器是一种神经网络架构，可将输入序列转换或更改为输出序列。对于 Amazon Personalize，输入序列是用户在您的数据中的物品交互历史记录。输出序列是他们的个性化推荐。有关转换器的更多信息，请参阅 AWS 云计算概念中心的什么是人工智能中的转换器？。

Personalized-Ranking-v2 使用与其他配方不同的定价模式。有关定价的更多信息，请参阅 Amazon Personalize 定价。

主题

- 配方特征
- 必需和可选数据集
- 属性和超参数

配方特征

Personalized-Ranking-v2 在对物品进行排名时使用以下 Amazon Personalize 配方特征：

- 实时个性化 - 使用实时个性化时，Amazon Personalize 会根据用户不断变化的兴趣来更新和调整物品推荐。有关更多信息，请参阅 实时个性化。
- 推荐元数据 - 使用 Personalized-Ranking-v2 配方时，如果您的物品数据集包含至少一列元数据，则市场活动可以自动选择在推荐结果中包括物品元数据。您没有手动为市场活动启用元数据。您可以使用元数据来丰富用户界面中的建议，例如将电影的类型添加到轮播中。有关更多信息，请参阅 推荐中的物品元数据。

必需和可选数据集

要使用 Personalized-Ranking-v2，您必须创建一个物品交互数据集，并导入至少 1000 个物品交互。Amazon Personalize 主要根据物品交互数据生成排名。有关更多信息，请参阅 物品交互数据。Personalized-Ranking-v2 可以对物品交互数据集和物品数据集中的多达 500 万个物品进行训练。

使用 Personalized-Ranking-v2 时，Amazon Personalize 可以使用包含以下内容的物品交互数据：

- 事件类型和事件值数据 - Amazon Personalize 使用事件类型数据（例如点击或观看事件类型），通过用户行为中的任何模式来识别用户的意图和兴趣。此外，您还可以在训练前使用事件类型和事件值数据来筛选记录。有关更多信息，请参阅 事件类型和事件值数据。

  > **Note**
  >
  > 使用 Personalized-Ranking-v2 时，在按事件类型或值进行筛选之前，训练费用将基于交互数据。有关定价的更多信息，请参阅 Amazon Personalize 定价。

- 上下文元数据 - 上下文元数据是您在事件发生时在用户环境中收集的交互数据，例如其位置或设备类型。有关更多信息，请参阅 上下文元数据。

以下数据集是可选的，可以用于改进推荐：

- 用户数据集 - Amazon Personalize 可以使用您的用户数据集中的数据来更好地了解您的用户及其兴趣。您还可以使用用户数据集中的数据来筛选推荐。有关您可以导入的用户数据的信息，请参阅用户元数据。
- 物品数据集 - Amazon Personalize 可以使用您的物品数据集中的数据来识别其行为中的关联和模式。这有助于 Amazon Personalize 了解您的用户及其兴趣。您还可以使用物品数据集中的数据来筛选推荐。有关您可以导入的物品数据的信息，请参阅物品元数据。

属性和超参数

Personalized-Ranking-v2 配方具有以下属性：

- 名称 – `aws-personalized-ranking-v2`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-personalized-ranking-v2`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-personalized-ranking-v2`

有关更多信息，请参阅 选择食谱。

下表描述了 Personalized-Ranking-v2 配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。为超参数选择最佳值的过程称为超参数优化 (HPO)。使用 Personalized-Ranking-v2 时，如果开启自动训练，则 Amazon Personalize 将每 90 天自动执行一次 HPO。如果没有开启自动训练，就不会执行 HPO。

该表为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）

名称	描述
算法超参数	
apply_recency_bias	确定模型是否应为物品交互数据集中的最新物品交互数据赋予更多权重。最新的交互数据可能包括交互事件底层模式的突然变化。 要训练一个使最近事件具有更高权重的模型，请将 apply_recency_bias 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 apply_recency_bias 设置为 false。 默认值：true 范围：true 或 false 值类型：布尔值 HPO 可调：否

Personalized-Ranking 食谱

> **Important**
>
> 我们建议使用 Personalized-Ranking-v2 配方。它可以通过更快的训练考虑多达 500 万个物品，并以更低的延迟生成更准确的排名。

Personalized-Ranking 食谱可生成物品的个性化排名。个性化排名 是为特定用户重新排名的推荐物品的列表。如果您有一系列已订购的物品（例如搜索结果、推广或精选列表），且想要为每位用户提供个

性化重新排名，则此食谱非常有用。例如，通过 Personalized-Ranking，Amazon Personalize 可以对您通过 OpenSearch 生成的搜索结果重新排名。

为了训练模型，Personalized-Ranking 配方使用物品交互数据集中的数据，以及（如果您创建了这些数据集）数据集组中的物品数据集和用户数据集（这些数据集是可选项）。使用 Personalized-Ranking，您的物品数据集可以包含非结构化文本元数据，物品交互数据集可以包含上下文元数据。要获取个性化排名，请使用 GetPersonalizedRanking API。

创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 Personalized-Ranking 时，您必须手动创建新的解决方案版本（重新训练模型），以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。然后，您必须使用解决方案版本更新所有市场活动。有关更多信息，请参阅 维护建议相关性。

> **Note**
>
> 如果您提供的物品没有用于排名的交互数据，则 Amazon Personalize 将在 GetPersonalizedRanking API 响应中返回这些没有建议分数的物品。

该配方具有以下属性：

- 名称 – `aws-personalized-ranking`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-personalized-ranking`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-personalized-ranking`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- 食谱类型 - `PERSONALIZED_RANKING`

超参数

下表描述了个性化排名配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]

- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与超级参数优化 (HPO)？

名称	描述
算法超参数	
hidden_dimension	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当物品交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。要确定最佳值，请使用 HPO。要使用 HPO，当您调用 performHPO 和 true 操作时，请将 CreateSolution 设置为 CreateSolutionVersion。 默认值：149 范围：[32, 256] 值类型：整数 HPO 可调：是
bptt	确定是否使用基于时间的反向传播技术。反向传播 是一种在基于递归神经网络的算法中更新权重的技术。将 bptt 用于长期积分以将延迟奖励与早期事件联系起来。例如，延迟奖励可以是几次点击后进行的购买。早期事件可以是初始点击。即使在相同的事件类型（例如点击）中，最好考虑长期影响并最大化总奖励。要考虑长期效果，请使用较大的 bptt 值。使用较大的 bptt 值需要更大的数据集和更多的处理时间。 默认值：32 范围：[2, 32] 值类型：整数 HPO 可调：是

名称	描述
recency_mask	确定模型是否应考虑物品交互数据集中的最新流行趋势。最新流行趋势可能包括交互事件的基本模式的突然变化。要训练一个使最近事件具有更高权重的模型，请将 recency_mask 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 recency_mask 设置为 false。要使用相等权重获得好的建议，您可能需要较大的训练数据集。 默认值：True 范围：True 或 False 值类型：布尔值 HPO 可调：是
特征化超参数	

名称	描述
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是有关用户的数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	描述
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是有关用户的数据总量。使用 max_user_history_length_percentile 排除具有较长历史记录长度的一定百分比的用户，因为这些用户的数据往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.99 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

Personalized-Ranking 示例笔记本

有关展示如何使用 Personalized-Ranking 食谱的示例 Jupyter 笔记本，请参阅 Personalize Ranking 示例。

语义相似度配方

语义相似度配方 (aws-semantic-similarity) 根据文本内容为语义上与给定项目相似的项目生成推荐。与依赖用户-项目交互的传统相似度配方不同，此配方分析物品的文本描述和属性，以生成嵌入并识别语义上相似的项目

此食谱非常适合您想要根据内容相似性推荐项目的场景，例如推荐主题相似的图书、相关主题的文章或描述相似的产品。它特别适用于交互历史记录有限的新项目（冷启动场景），以及语义关系比共现模式更重要的目录。

使用 Semantic-Similarity，您可以在GetRecommendations操作（或 Amazon Personalize 控制台）中提供商品编号，Amazon Personalize 会返回类似商品的清单。或者，您可以使用批量工作流程获取库存中所有商品的类似商品（请参阅获取批量商品推荐）。

主题

- 配方特征
- 必需和可选数据集
- 属性和超参数

配方特征

Semantic-Similarity 在生成商品推荐时使用以下 Amazon Personalize 食谱功能：

- 实时个性化 — 借助语义相似度配方，Amazon Personalize 会自动更新您的商品目录。当您向 Items 数据集添加新项目或更新现有项目元数据时，使用增量训练时，这些更改将在大约 30 分钟内反映在您的推荐中。这样可以确保您的客户始终看到您目录中最新的可用商品，而无需人工干预或等待完整的再培训周期。这对于经常变化的目录（例如新闻文章、博客文章或季节性产品）尤其有价值。要启用增量更新，客户必须：
  - 在 API 中将解决方案设置为 `performIncrementalUpdate` `true`
  - 在用户界面的 “训练方法” 下选择 “完整和增量训练” 或 “增量训练” 选项

  请注意，每当执行更新时，启用增量更新都会产生额外费用。
- 带有推荐的元数据 — 借助语义相似度配方，广告系列可以自动选择在推荐结果中包含商品元数据。您没有手动为市场活动启用元数据。您可以使用元数据来丰富用户界面中的建议，例如将电影的类型添加到轮播中。有关更多信息，请参阅推荐中的商品元数据。

必需和可选数据集

要使用语义相似度配方，必须创建 Items 数据集。Amazon Personalize 根据商品元数据的语义含义生成推荐。有关更多信息，请参阅项目元数据。语义相似度可以对项目数据集中多达 1000 万个项目进行训练。

借助语义相似性，Amazon Personalize 需要包含以下内容的商品数据

- itemName 字段-一个字符串字段，`itemName`设置为。`true`此字段应包含商品的标题或名称。
- 文本描述字段-至少一个标记为`textual`包含项目描述的字符串字段。这应该是最能描述和代表该项目的字段。

Amazon Personalize 使用此字段生成语义嵌入，以捕捉您的商品的含义和内容。

此外，如果您想使用基于新鲜度的排名，则应设置保留的 CREATION_TIMESTAMP 字段。有关更多信息，请参阅 属性和超参数。

以下数据集是可选的，可以用于改进推荐：

- 互动数据集 — Amazon Personalize 可以使用互动数据集中的数据，根据用户对项目的参与度计算人气分数。你可以使用人气分数根据类似物品在用户中的受欢迎程度对它们进行排名。如果要使用基于人气的排名，则必须提供互动数据集。您也可以使用互动数据集中的数据来筛选推荐。有关您可以导入的交互数据的信息，请参阅项目交互数据

属性和超参数

语义相似度配方具有以下特性：

- 名称 – `aws-semantic-similarity`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-semantic-similarity`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-semantic-similarity`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/aws-semantic-similarity`
- 食谱类型 - `RELATED_ITEMS`

有关更多信息，请参阅 选择食谱。

Semantic-Similarity 配方没有公开的超参数，但你可以在创建活动时配置受欢迎程度和新鲜度因子来影响相似物品的排名。

该表提供了每个因子的以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）

Name	说明
新鲜度	新鲜度系数表示一件物品的最新程度。新鲜度是通过根据物品的 CREATION_TIMESTAMP 对物品的使用寿命进行标准化来计算的。要使用新鲜度系数，您必须在 Items 数据集架构中包含 CREATION_TIMESTAMP 字段。较高的新鲜度系数值将在语义上相似的推荐中优先考虑较新的项目 默认值：`0.0` 射程：`[0.0, 1.0]` 值类型：双精度
人气	受欢迎程度系数根据用户互动来表示某件商品的受欢迎程度。受欢迎程度是通过标准化每件物品收到的互动次数来计算的。要使用人气因子，在创建数据集组时必须包含交互数据集。受欢迎程度因子值越高，在语义相似的推荐中，优先考虑与客户互动较多的项目。 默认值：`0.0` 射程：`[0.0, 1.0]` 值类型：双精度

请注意，新鲜度和受欢迎程度分数是在训练时计算的，增量更新不会更新受欢迎程度和新鲜度分数。要了解影响推荐项目排名的最新受欢迎程度和新鲜度因素，请使用自动再训练，或者手动重新训练解决方案，然后使用新的解决方案版本更新广告系列。

Similar-Items 食谱

> **Note**
>
> 所有 RELATED_ITEMS 食谱都使用交互数据。如果您还有物品元数据并希望 Amazon Personalize 使用它来查找类似物品，可选择 Similar-Items。或者，在您想为模型配置更多超参数时，选择 SIMS 配方。

Simalar-Item aws-similar-items s () 配方会为与您指定的物品相似的项目生成推荐。使用 Similar-Items，可帮助客户根据他们之前的行为和物品元数据发现您目录中的新物品。推荐类似物品可以提高您应用程序的用户参与度、点击率和转化率。

Similar-Items 根据交互数据和您提供的任何物品元数据计算相似度。它会考虑交互数据集用户历史记录中物品的共存情况，以及任何物品元数据的相似度。例如，对于 Similar-Items，Amazon Personalize 可以推荐客户经常购买的具有相似风格的物品（分类元数据），或者不同用户也观看过类似描述的电影（非结构化文本元数据）。

对于 Similar-Items，您可以在 GetRecommendations 操作（或 Amazon Personalize 控制台）中提供物品 ID，Amazon Personalize 会返回类似物品的列表。或者，您可以使用批量工作流程针对库存中的所有物品获取相似物品（请参阅获取批量物品推荐）。当您获得类似物品时，您可以根据您在请求中指定的物品的属性筛选物品。为此，请将 `CurrentItem.attribute` 元素添加到您的筛选器。有关示例，请参阅item data filter examples。

要使用 Similar-Items，您必须创建一个包含至少 1000 个唯一历史和事件交互（组合）的物品交互数据集。为了获得更准确的预测，我们建议您同时创建物品数据集并导入有关目录中物品的元数据。Similar-Items 在生成建议时不使用用户数据集中的数据。您仍然可以根据用户数据集中的数据筛选建议。有关更多信息，请参阅 筛选建议和用户细分。

如果您的物品数据集包含文本数据和物品标题数据，则可以在批量建议中为相关物品生成主题。有关更多信息，请参阅 内容生成器中带有主题的批量建议。

您可以获得与冷物品（交互次数少于五次的物品）相似的物品的建议。如果 Amazon Personalize 找不到您在建议请求或批量输入文件中指定的物品 ID，则食谱会将热门物品作为建议返回。

创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 Similar-Items 时，您必须手动创建新的解决方案版本（重新训练模型），以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。然后，您必须使用解决方案版本更新所有市场活动。有关更多信息，请参阅 维护建议相关性。

属性和超参数

Similar-Items 食谱具有以下属性：

- 名称 – `aws-similar-items`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-similar-items`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-similar-items`

有关更多信息，请参阅 选择食谱。

下表描述 Similar-Items 食谱的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO？

名称	说明
算法超参数	
`popularity_discount_factor`	配置受欢迎程度如何影响建议。指定一个接近 0 的值可包含更受欢迎的物品。指定一个更接近 1 的值可减少对受欢迎程度的关注。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否
`item_id_hidden_dim`	Amazon Personalize 根据交互数据对物品 ID 嵌入进行建模时使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。要使用 `item_id_hidden_dim` ，您必须使用 HPO 并提供最小和最大范围值。Amazon Personalize 使用 HPO 来查找您指定范围内的最佳值。当物品交互数据集较大时，指定更大的最大值。使用较大的最大值需要更多处理时间。 要使用 HPO，请在您调用 CreateSolution 操作时，将 `performHPO` 设置为 `true`。 默认值：100

名称	说明
	范围：[30、200] 值类型：整数 HPO 可调：是
item_metadata_hidden_dim	Amazon Personalize 用来对物品元数据建模的隐藏变量的数量。要使用 item_metadata_hidden_dim ，您必须使用 HPO 并提供最小和最大范围值。Amazon Personalize 使用 HPO 来查找您指定范围内的最佳值。当物品交互数据集较大时，指定更大的最大值。使用较大的最大值需要更多处理时间。 要使用 HPO，请在您调用 CreateSolution 操作时，将 performHPO 设置为 true。 默认值：100 范围：[30、200] 值类型：整数 HPO 可调：是

SIMS 配方

> **Note**
>
> 所有 RELATED_ITEMS 食谱都使用交互数据。在您想为模型配置更多超参数时，选择 SIMS。如果您有物品元数据并希望 Amazon Personalize 使用它来查找类似物品，可选择 Similar-Items 食谱。

Item-to-item相似之处 (SIMS) 配方使用协作筛选来推荐与你获得推荐时指定的项目最相似的项目。SIMS 使用您的物品交互数据集（而不是颜色或价格等物品元数据）来确定相似度。SIMS 会识别交互数据集用户历史记录中物品的共存情况，以推荐相似物品。例如，借助 SIMS，Amazon Personalize 可以推荐客户经常一起购买的咖啡店物品或不同用户也观看的电影。

当您获得类似物品建议时，您可以根据您在请求中指定的物品的属性筛选物品。为此，请将 `CurrentItem.attribute` 元素添加到您的筛选器。有关示例，请参阅item data filter examples。

要使用 SIMS，您必须创建一个包含至少 1000 个唯一历史和事件交互（组合）的物品交互数据集。SIMS 在生成建议时不使用用户或物品数据集中的数据。您仍然可以根据这些数据集中的数据筛选建议。有关更多信息，请参阅 筛选建议和用户细分。

如果某件物品的用户行为数据不足，或者找不到您提供的物品 ID，则 SIMS 会推荐热门物品。创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 SIMS 时，您必须手动创建新的解决方案版本（重新训练模型），以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。然后，您必须使用解决方案版本更新所有市场活动。有关更多信息，请参阅 维护建议相关性。

SIMS 食谱具有以下属性：

- 名称 – `aws-sims`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-sims`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-sims`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/sims`
- 食谱类型 - `RELATED_ITEMS`

下表描述 SIMS 配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与超级参数优化 (HPO)？

Name	说明
算法超参数	

Name	说明
popularity_discount_factor	配置受欢迎程度如何影响建议。指定一个接近 0 的值可包含更受欢迎的物品。指定一个更接近 1 的值可减少对受欢迎程度的关注。 默认值：0.5 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：是
min_cointeraction_count	计算一对物品之间的相似性所需的最小共同交互次数。例如，值 3 表示您需要三个或更多与这两个物品进行交互的用户以供算法计算其相似性。 默认值：3 范围：[0, 10] 值类型：整数 HPO 可调：是
特征化超参数	

Name	说明
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是用户的可用数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 默认值：0.005 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

Name	说明
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是用户的可用数据总量。使用 max_user_history_length_percentile 排除历史记录长度较长的一定百分比的用户。具有较长历史记录的用户往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，min_hist_length_percentile = 0.05 和 max_hist_length_percentile = 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.995 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

Name	说明
min_item_interaction_count_percentile	要包含在模型训练中的物品交互计数的最小百分位数。使用 min_item_interaction_count_percentile 排除交互历史记录较短的百分比的物品。历史记录较短的物品通常是新物品。删除它们可以在训练模型时更多地关注具有已知历史记录的物品。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数物品，但会删除边缘案例。 默认值：0.01 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

Name	说明
max_item_interaction_count_percentile	要包含在模型训练中的物品交互计数的最大百分位数。使用 max_item_interaction_count_percentile 排除交互历史记录较长的百分比的物品。具有较长的历史记录的物品往往较旧，可能已过时。例如，已绝版的电影发行版本。删除这些物品可以专注于相关性更高的物品。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数物品，但会删除边缘案例。 例如，min_item_interaction_count_percentile = 0.05 和 max_item_interaction_count_percentile = 0.95 将包括除交互计数在最低或最高 5% 内的物品之外的所有物品。 默认值：0.9 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

SIMS 示例笔记本

有关演示如何使用 SIMS 配方的 Jupyter 笔记本示例，请参阅查找类似物品 + HPO。

Next-Best-Action 配方

Next-Best-Action (aws-next-best-action) 配方为您的用户生成下一个最佳操作的实时建议。用户的下一个最佳操作是他们最有可能采取的操作。例如，注册您的忠诚度计划、下载您的应用或申请信用卡。

借助 Next-Best-Action，您可以在用户使用您的应用程序时为他们提供个性化的操作建议。为用户建议正确的操作可能会导致更多用户采取您的操作。根据您要建议的操作，您可以提高客户忠诚度、创造更多收入并改善应用程序的用户体验。有关描述个性化操作建议如何有利于电子商务应用程序的使用案例示例，请参阅使用案例示例。

Amazon Personalize 会根据您导入到操作数据集的操作预测下一个最佳操作。它根据用户与操作和物品的交互来确定他们最可能采取的操作。如果您的操作数据包含操作的值，则 Amazon Personalize 会将该操作的值计算在内。如果用户执行两个不同操作的可能性相同，则 Amazon Personalize 会使值越高的操作排名越高。

当您收到用户的实时操作建议时，Amazon Personalize 会返回用户最有可能在可配置的时间段内执行的操作列表 (`action optimization period`)。例如，他们在接下来的 14 天内最有可能采取的操作。该列表按倾向分数以降序排序。该分数表示用户将执行此操作的可能性。

在您导入操作交互数据之前，Amazon Personalize 会在您的操作数据集中推荐不进行个性化的操作，倾向分数为 0.0。在操作达到以下条件后，该操作将获得分数：

- 至少 50 次 TAKEN 事件类型的操作交互。
- 至少 50 次 NOT_TAKEN 或 VIEWED 事件类型的操作交互。

这些操作交互必须出现在最新的解决方案版本训练中，并且必须在操作交互数据集中的最新交互时间戳后的 6 周内发生。

有关 Next-Best-Action 配方使用的数据的更多信息，请参阅必需和可选数据集。

当您使用 Next-Best-Action 配方创建解决方案时，您可以使用 `action optimization period` 特征化超参数配置 Amazon Persalization 在预测操作时使用的时间窗口。有关更多信息，请参阅 属性和超参数。

主题

- 使用案例示例
- 配方特征
- 必需和可选数据集
- 属性和超参数

使用案例示例

为用户建议正确的操作可能会导致更多用户采取您的操作。根据您要建议的操作，您可以潜在提高客户忠诚度、创造更多收入并改善应用程序的用户体验。

例如，您可能有一个电子商务应用程序建议执行以下不同操作：

- 订阅忠诚度计划

- 下载移动应用
- 在珠宝类别中购买
- 在美容和美发类别中购买

您的用户可能经常在您的网站上购物，并且反复执行珠宝以及美容和美发购买操作。对于该用户，Amazon Personalize 操作建议及其分数可能包括以下内容：

- 订阅忠诚度计划

  倾向分数 – 1.00
- 在珠宝类别中购买

  倾向分数 – 0.86
- 在美容和美发类别中购买

  倾向分数 – 0.85

有了这些操作建议，您就知道要提示用户注册您的忠诚度计划。此操作的倾向分数最高，也是用户最有可能采取的操作。这是因为用户经常在您的商店购物，并且很可能会从您的忠诚度计划中获得好处。

配方特征

Next-Best-Action 配方在生成操作建议时使用以下 Amazon Personalize 配方特征：

- 实时个性化：Amazon Personalize 使用实时个性化功能，根据用户不断变化的兴趣来更新和调整操作建议。有关更多信息，请参阅 实时个性化。
- 浏览：通过浏览，建议包括新的操作或交互数据较少的操作。有关浏览的更多信息，请参阅探索。
- 自动更新：通过自动更新，Amazon Personalize 每两小时自动更新一次最新模型（解决方案版本），以便通过浏览将新操作纳入到建议中。有关更多信息，请参阅 自动更新。

必需和可选数据集

要使用 Next-Best-Action 配方，必须创建以下数据集：

- 操作：您将有关您的操作的数据（例如其值）导入到 Amazon Personalize 操作数据集中。

  在您的操作数据中，您可以为每个操作提供一个 EXPIRATION_TIMESTAMP。如果某操作已过期，Amazon Personalize 将不会将其包含在建议中。您也可以为每个操作提供

REPEAT_FREQUENCY。这表示在用户与操作交互后，Amazon Personalize 应等待多长时间才能再次建议该操作。有关操作数据集可以存储的数据的信息，请参阅操作元数据。

- 物品交互：您的物品交互数据集必须至少有 1000 次物品交互。Amazon Personalize 使用物品交互来了解用户的当前状态及其兴趣。有关物品交互数据的信息，请参阅物品交互数据。

以下数据集是可选的：

- 操作交互数据集：操作交互是指涉及用户和操作数据集中的操作的交互。您可以导入“已执行”、“未执行”和“已查看”的操作交互。尽管此数据是可选的，但我们建议您导入操作交互数据以获得高质量的建议。如果您没有操作交互数据，则可以创建一个空的操作交互数据集，并使用 PutActionInteractions API 操作记录客户与操作的交互。

  在您导入操作交互数据之前，Amazon Personalize 会在您的操作数据集中推荐不进行个性化的操作，倾向分数为 0.0。在操作达到以下条件后，该操作将获得分数：

  - 至少 50 次 TAKEN 事件类型的操作交互。
  - 至少 50 次 NOT_TAKEN 或 VIEWED 事件类型的操作交互。

  这些操作交互必须出现在最新的解决方案版本训练中，并且必须在操作交互数据集中的最新交互时间戳后的 6 周内发生。

  有关您可以导入的操作交互数据的信息，请参阅操作交互数据。有关记录操作交互事件的信息，请参阅记录实时操作交互事件。

  > **Note**
  >
  > 使用 Next-Best-Action 时，Amazon Personalize 不会在操作交互数据集中使用展示数据或上下文元数据。

- 用户：Amazon Personalize 使用您的用户数据集中的任何数据来更好地了解您的用户及其兴趣。您还可以使用用户数据集中的数据来筛选操作建议。有关您可以导入的用户数据的信息，请参阅用户元数据。
- 物品：Amazon Personalize 使用您的物品数据集以及您的物品交互数据集中的任何数据来识别其行为中的关联和模式。这有助于 Amazon Personalize 了解您的用户及其兴趣。有关您可以导入的物品数据的信息，请参阅物品元数据。

属性和超参数

Next-Best-Action 配方不支持超参数优化。Next-Best-Action 配方具有以下属性：

- 名称 – `aws-next-best-action`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-next-best-action`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-next-best-action`

下表描述了 aws-next-best-action 配方的特征化超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。特征化超参数控制如何筛选训练中使用的数据。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO

名称	描述
特征化超参数	
`action_optimization_period`	Amazon Personalize 在预测用户的下一个最佳操作时使用的时间窗口。例如，用户在接下来的 14 天内最有可能采取的操作。 如果您没有太多操作交互数据，请指定较大的值。如果您不确定要指定什么值，请使用默认值。 默认值：14 范围：[7、28] 值类型：整数 HPO 可调：否

Item-Affinity 食谱

Item-Affinity (aws-item-affinity) 食谱是一个 USER_SEGMENTATION 食谱，可为您指定的每个物品创建一个用户细分（用户组）。Amazon Personalize 预测这些用户最有可能与每件物品交互。使用 Item-Affinity 进一步了解您的用户，并根据他们各自的用户细分执行操作。

例如，您可能想根据用户对目录中物品的偏好为零售应用程序创建市场活动。Item-Affinity 将根据交互和物品数据集中的数据为每个物品创建一个用户细分。您可以使用此类细分，根据不同的用户细分执行操作（例如，单击物品或购买物品）的可能性向这些细分推广不同的物品。其他用途可能包括向不同的用户群交叉销售产品或识别潜在的求职者。

要根据物品获取用户细分，您需使用 Item-Affinity 食谱创建解决方案和解决方案版本，然后将 JSON 格式的物品列表添加到 Amazon S3 存储桶并创建批量细分作业。Amazon Personalize 将每件物品的用户细分输出到您在 Amazon S3 中的输出位置。您的输入数据最多可有 500 个用于获取用户细分的物品。有关为批量细分作业准备输入数据的信息，请参阅为批量建议准备输入数据。

您必须拥有物品交互数据集才能使用 Item-Affinity。物品和用户数据集是可选项。您可以使用批量细分作业来获取用户细分。有关更多信息，请参阅 获取批量用户细分。

创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 Item-Affinity 时，您必须创建新的解决方案版本，以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。要获取物品的用户细分，物品必须在您创建解决方案版本时存在。

Item-Affinity 食谱具有以下属性：

- 名称 – `aws-item-affinity`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-item-affinity`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-item-affinity`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/item-affinity`
- 食谱类型 - `USER_SEGMENTATION`

下表描述 Item-Affinity 食谱的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。您不能在 Item-Affinity 食谱中使用超参数优化 (HPO)。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）

名称	描述
算法超参数	
hidden_dimension	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。 默认值：149 范围：[32, 256] 值类型：整数

Item-Attribute-Affinity 食谱

Item-Attribute-Affinity (aws-item-attribute-affinity) 食谱是一个 USER_SEGMENTATION 食谱，可为您指定的每个物品属性创建一个用户细分（用户组）。Amazon Personalize 预测这些用户最有可能与物品特定属性交互。使用 Item-Attribute-Affinity 进一步了解您的用户，并根据他们各自的用户细分执行操作。

例如，您可能想根据用户对目录中鞋子类型的偏好为零售应用程序创建市场活动。Item-Attribute-Affinity 将根据交互和物品数据集中的数据为每个类型的鞋子创建一个用户细分。您可以使用此类细分，根据不同的用户细分执行操作（例如，单击鞋子或购买鞋子）的可能性向这些细分推广不同的鞋子。其他用途可能包括向不同的用户推广不同的电影类型，或者根据工作类型识别潜在的求职者。

要根据物品属性获取用户细分，您需使用 Item-Attribute-Affinity 食谱创建解决方案和解决方案版本，然后将 JSON 格式的物品属性列表添加到 Amazon S3 存储桶并创建批量细分作业。Amazon Personalize 将每件物品的用户细分输出到您在 Amazon S3 中的输出位置。您的输入数据最多可以有 10 个查询，其中每个查询都是一个或多个物品属性。有关为批量细分作业准备输入数据的信息，请参阅为批量建议准备输入数据。

您必须拥有物品交互数据集和物品数据集才能使用 Item-Attribute-Affinity。您的物品数据集必须至少有一列是非文本、非保留的元数据列。您可以使用批量细分作业来获取用户细分。有关更多信息，请参阅使用自定义资源获取批量用户细分。

创建解决方案版本后，确保您的解决方案版本和数据为最新状态。使用 Item-Attribute-Affinity 时，您必须创建新的解决方案版本，以便 Amazon Personalize 将新用户纳入用户细分，并更新模型以反映用户的最新行为。要获取物品属性的用户细分，物品属性必须在您创建解决方案版本时存在。

Item-Attribute-Affinity 食谱具有以下属性：

- 名称 – `aws-item-attribute-affinity`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-item-attribute-affinity`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-item-attribute-affinity`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/item-attribute-affinity`
- 食谱类型 - `USER_SEGMENTATION`

下表描述 Item-Attribute-Affinity 食谱的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。您不能在 Item-Attribute-Affinity 食谱中使用超参数优化 (HPO)。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）

名称	描述
算法超参数	
`hidden_dimension`	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。 默认值：149 范围：[32, 256]

名称	描述
	值类型：整数

旧版 HRNN 配方

旧版 HRNN 食谱不再可用。此文档仅供参考。

我们建议使用 aws-user-personalizaton（用户个性化）配方，而不是传统的 HRNN 配方。User-Personalization 改进并统一了 HRNN 食谱提供的功能。有关更多信息，请参阅 User-Personalization 食谱。

Amazon Personalize 可以根据其输入数据的分析自动选择最合适的分层递归神经网络 (HRNN) 食谱。该选项称为 AutoML。有关更多信息，请参阅 使用 AutoML 选择 HRNN 食谱（仅限 API）。

主题

- 使用 AutoML 选择 HRNN 食谱（仅限 API）
- HRNN 食谱（旧版）
- HRNN-Metadata 食谱（旧版）
- HRNN-Coldstart 食谱（旧版）

使用 AutoML 选择 HRNN 食谱（仅限 API）

Amazon Personalize 可以根据其输入数据的分析自动选择最合适的分层递归神经网络 (HRNN) 食谱。该选项称为 AutoML。要执行 AutoML，请在调用 CreateSolution API 时将 `performAutoML` 参数设置为 `true`。

您还可以指定 Amazon Personalize 检查的食谱列表，以根据您指定的指标决定最佳食谱。在这种情况下，您调用 `CreateSolution` 操作，为 `performAutoML` 参数指定 `true`，忽略 `recipeArn` 参数，包含 `solutionConfig` 参数，并指定 `metricName` 和 `recipeList` 作为 `autoMLConfig` 对象的一部分。

下表显示了选择配方的方式。必须指定 `performAutoML` 或 `recipeArn`，但不能同时指定二者。仅使用 HRNN 配方执行 AutoML。

performAutoML	recipeArn	solutionConfig	结果
true	忽略	已忽略	Amazon Personalize 选择食谱
true	忽略	`autoMLConfig` ：指定了 `metricName` 和 `recipeList`	Amazon Personalize 从列表中选择优化指标的食谱
忽略	满足指定	已忽略	您指定配方
忽略	满足指定	满足指定	您指定配方并覆盖默认训练属性

> **Note**
>
> 当 `performAutoML` 为 `true` 时，将忽略 `solutionConfig` 对象的所有参数，`autoMLConfig` 除外。

HRNN 食谱（旧版）

> **Note**
>
> 旧版 HRNN 食谱不再可用。此文档仅供参考。
> 我们建议使用 aws-user-personalization (User-Personalization) 食谱，而不是旧版 HRNN 食谱。User-Personalization 改进并统一了 HRNN 食谱提供的功能。有关更多信息，请参阅 User-Personalization 食谱。

Amazon Personalize 分层递归神经网络 (HRNN) 食谱模型改变用户行为以在会话期间提供建议。会话是给定时间范围内的一组用户交互，例如，其目标是找到满足需求的特定物品。通过将用户最近交互的权重提高，您可以在会话期间提供更相关的建议。

HRNN 适用于可随时间变化的用户意图和兴趣。它获取有序的用户历史记录并自动对它们进行加权以做出更好的推论。HRNN 使用门控机制将折扣权重建模为物品和时间戳的可学习函数。

Amazon Personalize 从数据集中为每个用户派生特征。如果您已完成实时数据集成，则会根据用户活动实时更新这些特征。要获取建议，您只需提供 USER_ID。如果您还提供了 ITEM_ID，则 Amazon Personalize 会将其忽略。

HRNN 配方具有以下属性：

- 名称 – aws-hrnn
- 食谱 Amazon 资源名称 (ARN) - arn:aws:personalize:::recipe/aws-hrnn
- 算法 ARN - arn:aws:personalize:::algorithm/aws-hrnn
- 特征转换 ARN - arn:aws:personalize:::feature-transformation/JSON-percentile-filtering
- 食谱类型 - USER_PERSONALIZATION

下表描述 HRNN 配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO？

名称	描述
算法超参数	
hidden_dimension	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当物品交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。要确定最佳值，请使用 HPO。要使用 HPO，当您调用 performHPO 和 true 操作时，请将 CreateSolution 设置为 CreateSolutionVersion。 默认值：43

名称	描述
	范围：[32, 256] 值类型：整数 HPO 可调：是
bptt	确定是否使用基于时间的反向传播技术。反向传播 是一种在基于递归神经网络的算法中更新权重的技术。将 bptt 用于长期积分以将延迟奖励与早期事件联系起来。例如，延迟奖励可以是几次点击后进行的购买。早期事件可以是初始点击。即使在相同的事件类型（例如点击）中，最好考虑长期影响并最大化总奖励。要考虑长期效果，请使用较大的 bptt 值。使用较大的 bptt 值需要更大的数据集和更多的处理时间。 默认值：32 范围：[2, 32] 值类型：整数 HPO 可调：是
recency_mask	确定模型是否应考虑物品交互数据集中的最新流行趋势。最新流行趋势可能包括交互事件的基本模式的突然变化。要训练一个使最近事件具有更高权重的模型，请将 recency_mask 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 recency_mask 设置为 false。要使用相等权重获得好的建议，您可能需要较大的训练数据集。 默认值：True 范围：True 或 False 值类型：布尔值 HPO 可调：是

名称	描述
特征化超参数	
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是有关用户的数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	描述
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是有关用户的数据总量。使用 max_user_history_length_percentile 排除具有较长历史记录长度的一定百分比的用户，因为这些用户的数据往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.99 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

HRNN-Metadata 食谱（旧版）

> **Note**
>
> 旧版 HRNN 食谱不再可用。此文档仅供参考。
> 我们建议使用 aws-user-personalization (User-Personalization) 食谱，而不是旧版 HRNN 食谱。User-Personalization 改进并统一了 HRNN 食谱提供的功能。有关更多信息，请参阅 User-Personalization 食谱。

HRNN 元数据配方可预测用户将与之交互的物品。它与具有从上下文、用户和物品元数据派生的附加功能（分别来自交互、用户和物品数据集）的 HRNN 配方相似。当有高质量的元数据可用时，HRNN 元数据可提供高于非元数据模型的准确性。使用该配方可能需要更长的训练时间。

HRNN 元数据配方具有以下属性：

- 名称 – `aws-hrnn-metadata`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-hrnn-metadata`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-hrnn-metadata`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/featurize_metadata`
- 食谱类型 - `USER_PERSONALIZATION`

下表描述 HRNN 元数据配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与超级参数优化 (HPO)？

名称	描述
算法超参数	
`hidden_dimension`	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当物品交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。要确定最佳值，请使用 HPO。要使用 HPO，当您调用 `performHPO` 和 `true` 操作时，请将 CreateSolution 设置为 CreateSolutionVersion。 默认值：43 范围：[32, 256] 值类型：整数

名称	描述
	HPO 可调：是
bptt	确定是否使用基于时间的反向传播技术。反向传播 是一种在基于递归神经网络的算法中更新权重的技术。将 bptt 用于长期积分以将延迟奖励与早期事件联系起来。例如，延迟奖励可以是几次点击后进行的购买。早期事件可以是初始点击。即使在相同的事件类型（例如点击）中，最好考虑长期影响并最大化总奖励。要考虑长期效果，请使用较大的 bptt 值。使用较大的 bptt 值需要更大的数据集和更多的处理时间。 默认值：32 范围：[2, 32] 值类型：整数 HPO 可调：是
recency_mask	确定模型是否应考虑物品交互数据集中的最新流行趋势。最新流行趋势可能包括交互事件的基本模式的突然变化。要训练一个使最近事件具有更高权重的模型，请将 recency_mask 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 recency_mask 设置为 false。要使用相等权重获得好的建议，您可能需要较大的训练数据集。 默认值：True 范围：True 或 False 值类型：布尔值 HPO 可调：是
特征化超参数	

名称	描述
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是有关用户的数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	描述
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是有关用户的数据总量。使用 max_user_history_length_percentile 排除具有较长历史记录长度的一定百分比的用户，因为这些用户的数据往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.99 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

HRNN-Coldstart 食谱（旧版）

> **Note**
>
> 旧版 HRNN 食谱不再可用。此文档仅供参考。
> 我们建议使用 aws-user-personalization (User-Personalization) 食谱，而不是旧版 HRNN 食谱。User-Personalization 改进并统一了 HRNN 食谱提供的功能。有关更多信息，请参阅 User-Personalization 食谱。

当您经常添加新物品和交互并希望立即获得有关这些物品的建议时，可以使用 HRNN-Coldstart 食谱来预测用户将与之交互的物品。HRNN 冷启动配方类似于 HRNN 元数据配方，但它允许您获取有关新物品的建议。

此外，当您由于最近的流行趋势或由于交互可能极不寻常并且在训练中引入噪音而希望从训练物品中排除具有长交互列表的物品时，也可以使用 HRNN 冷启动配方。使用 HRNN 冷启动，您可以筛选掉相关性较弱的物品以创建用于训练的子集。该物品子集称为冷物品，是在物品交互数据集中具有相关交互事件的物品。如果物品符合以下条件，则该物品被视为冷物品：

- 交互数量少于指定的最大交互数量。您在配方的 `cold_start_max_interactions` 超参数中指定此值。
- 相对持续时间短于最大持续时间。您在配方的 `cold_start_max_duration` 超参数中指定此值。

要减少冷物品的数量，请为 `cold_start_max_interactions` 或 `cold_start_max_duration` 设置较低的值。要增加冷物品的数量，请为 `cold_start_max_interactions` 或 `cold_start_max_duration` 设置较大的值。

HRNN-Coldstart 具有以下冷物品限制：

- `Maximum cold start items`：80,000
- `Minimum cold start items`：100

如果冷物品的数量超出此范围，则尝试创建解决方案将失败。

HRNN-Coldstart 配方具有以下属性：

- 名称 – `aws-hrnn-coldstart`
- 食谱 Amazon 资源名称 (ARN) - `arn:aws:personalize:::recipe/aws-hrnn-coldstart`
- 算法 ARN - `arn:aws:personalize:::algorithm/aws-hrnn-coldstart`
- 特征转换 ARN - `arn:aws:personalize:::feature-transformation/featurize_coldstart`
- 食谱类型 - `USER_PERSONALIZATION`

有关更多信息，请参阅 选择食谱。

下表描述 HRNN-Coldstart 配方的超参数。超参数 是一个算法参数，您可以调整该参数以提高模型性能。算法超参数控制模型的执行方式。特征化超参数控制如何筛选训练中使用的数据。为超参数选择最佳值的过程称为超参数优化 (HPO)。有关更多信息，请参阅 超级参数和 HPO。

该表还为每个超参数提供以下信息：

- 范围：[上界, 下界]
- 值类型：Integer、Continuous（浮点数）、Categorical（布尔值、列表、字符串）
- 可调 HPO：该参数是否可以参与 HPO？

名称	描述
算法超参数	
hidden_dimension	模型中使用的隐藏变量的数量。隐藏变量 重新创建用户的购买历史记录和物品统计数据来生成排名分数。当物品交互数据集包含更复杂的模式时，请指定更多数量的隐藏维度。使用更多隐藏维度需要更大的数据集和更多的处理时间。要确定最佳值，请使用 HPO。要使用 HPO，当您调用 performHPO 和 true 操作时，请将 CreateSolution 设置为 CreateSolutionVersion。 默认值：149 范围：[32, 256] 值类型：整数 HPO 可调：是
bptt	确定是否使用基于时间的反向传播技术。反向传播 是一种在基于递归神经网络的算法中更新权重的技术。将 bptt 用于长期积分以将延迟奖励与早期事件联系起来。例如，延迟奖励可以是几次点击后进行的购买。早期事件可以是初始点击。即使在相同的事件类型（例如点击）中，最好考虑长期影响并最大化总奖励。要考虑长期效果，请使用较大的 bptt 值。使用较大的 bptt 值需要更大的数据集和更多的处理时间。 默认值：32 范围：[2, 32] 值类型：整数

名称	描述
	HPO 可调：是
recency_mask	确定模型是否应考虑物品交互数据集中的最新流行趋势。最新流行趋势可能包括交互事件的基本模式的突然变化。要训练一个使最近事件具有更高权重的模型，请将 recency_mask 设置为 true。要训练一个使所有过往交互占相同权重的模型，请将 recency_mask 设置为 false。要使用相等权重获得好的建议，您可能需要较大的训练数据集。 默认值：True 范围：True 或 False 值类型：布尔值 HPO 可调：是
特征化超参数	
cold_start_max_interactions	物品可能必须被视为冷物品的最大用户-物品交互次数。 默认值：15 范围：正整数 值类型：整数 HPO 可调：否

名称	描述
cold_start_max_duration	用户-物品交互被视为冷启动物品的相对于起点的最大持续时间（以天为单位）。要设置用户-物品交互的起点，请设置 cold_start_relative_from 超参数。 默认值：5.0 范围：正浮点数 值类型：浮点数 HPO 可调：否
cold_start_relative_from	确定 HRNN 冷启动配方计算 cold_start_max_duration 的起点。要从当前时间计算，请选择 currentTime 。 要从物品交互数据集中最新物品的时间戳计算 cold_start_max_duration ，请选择 latestItem 。如果您经常添加新物品，该设置将非常有用。 默认值：latestItem 范围：currentTime 、latestItem 值类型：字符串 HPO 可调：否

名称	描述
min_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最小百分位数。历史记录长度 是有关用户的数据总量。使用 min_user_history_length_percentile 排除历史记录长度较短的一定百分比的用户。历史记录较短的用户通常会根据物品受欢迎程度而不是用户的个人需求或需要来显示模式。删除它们可以在训练模型时更多地关注数据中的基础模式。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.0 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

名称	描述
max_user_history_length_percentile	要包含在模型训练中的用户历史记录长度的最大百分位数。历史记录长度 是有关用户的数据总量。使用 max_user_history_length_percentile 排除具有较长历史记录长度的一定百分比的用户，因为这些用户的数据往往包含噪音。例如，机器人可能有很长的自动交互列表。删除这些用户可限制训练中的噪音。使用直方图或类似工具查看用户历史记录长度后，请选择适当的值。我们建议您设置一个值，该值将保留大多数用户，但会删除边缘案例。 例如，设置 min__user_history_length_percentile to 0.05 和 max_user_history_length_percentile to 0.95 将包括除那些历史记录长度在最低或最高 5% 内的用户之外的所有用户。 默认值：0.99 范围：[0.0, 1.0] 值类型：浮点数 HPO 可调：否

为 Amazon Personalize 准备训练数据

选择域应用场景或配方并记下其数据要求后，就可以开始准备数据了。Amazon Personalize 可以使用以下数据类型：

- 物品交互 - 在 Amazon Personalize 中，物品交互是用户与您目录中的物品之间的积极交互事件。例如，用户在看电影、查看商品列表或购买一双鞋。
- 物品 - 物品元数据可能包括目录中每个物品的价格、SKU 类型、描述或库存等信息。
- 用户 - 用户元数据可能包括每个用户的年龄、性别、会员资格和兴趣等信息。
- 操作 - 操作是您可能需要向客户推荐的交互活动。操作可能包括安装您的移动应用、填写会员资料、加入忠诚度计划或注册促销电子邮件。对于 Next-Best-Action 配方，需要操作数据集。没有其它自定义配方或域使用案例使用操作数据。
- 操作交互 - 操作交互是用户和操作之间的交互事件。Next-Best-Action 配方使用这些数据和操作数据集中的数据向用户建议操作。没有其他自定义配方或域应用场景使用操作交互数据。

Amazon Personalize 将数据存储在数据集中，每种数据类型对应一个数据集。每个数据集都有不同的要求。将数据导入 Amazon Personalize 数据集时，您可以选择批量、单独导入记录（或两者兼而有之）。批量导入涉及导入 Amazon S3 存储桶中的一个或多个 CSV 文件中存储的大量历史记录。

- 如果您没有批量数据，则可以使用单独的导入操作来收集数据和直播事件，直到满足 Amazon Personalize 训练要求以及域应用场景或配方的数据要求。有关记录事件的信息，请参阅记录实时事件以影响推荐。有关导入单个记录的信息，请参阅将单个记录导入 Amazon Personalize 数据集。
- 如果您不确定自己是否有足够的数据或对数据质量存有疑问，则可以将您的数据导入 Amazon Personalize 数据集，然后使用 Amazon Personalize 对其进行分析。有关更多信息，请参阅 分析 Amazon Personalize 数据集中的数据质量和数量。

以下各节介绍了每种 Amazon Personalize 数据集类型的数据要求和准备批量数据的准则。如果您没有批量数据，请查看各节以了解可以通过单个导入操作导入的必需数据和可选数据。如果您在设置数据格式方面需要其他帮助，可以使用 Amazon SageMaker AI Data Wrangler（Data Wrangler）来准备数据。有关更多信息，请参阅 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

主题

- 适用于所有数据类型的批量数据格式准则
- 为训练准备物品作交互数据
- 为训练准备物品元数据
- 为训练准备用户元数据
- 为训练准备操作元数据
- 为训练准备操作交互数据

适用于所有数据类型的批量数据格式准则

以下指南和要求有助于确保批量数据格式正确。

- 您的输入数据必须在 CSV（逗号分隔值）文件中。
- CSV 文件的第一行必须包含列标题。不要将标题括在引号 (") 中。
- 列必须具有唯一的字母数字名称。例如，您不能同时添加 `GENRES_FIELD_1` 字段和 `GENRESFIELD1` 字段。
- 如果您要导入多个 CSV 文件，则所有文件的列标题必须完全一致。
- 确保您的数据集类型有必填字段，并确保其名称符合 Amazon Personalize 要求。例如，您的物品数据可能有一个名为 `ITEM_IDENTIFICATION_NUMBER` 的列，其中包含每件物品的 ID。要将此列用作 ITEM_ID 字段，请将该列重命名为 `ITEM_ID`。如果您使用 Data Wrangler 来设置数据格式，则可以使用为 Amazon Personalize 映射列 Data Wrangler 转换，来确保您的列命名正确。

  有关使用 Data Wrangler 准备数据的信息，请参阅使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。
- CSV 文件中的每条记录必须都在单行中。
- Amazon Personalize 不支持复杂数据类型，例如数组和映射。
- 要让 Amazon Personalize 在训练或筛选时使用布尔数据，请使用字符串值 `"True"` 或数字值 `1` 代表 true，使用字符串值 `"False"` 或数字值 `0` 代表 false。
- 如果您使用 Data Wrangler 设置数据格式，则可以使用 Data Wrangler 转换将值解析为类型来转换数据类型。
- `TIMESTAMP` 和 `CREATION_TIMESTAMP` 数据必须采用 UNIX 纪元 时间格式。有关更多信息，请参阅 时间戳数据。
- 避免在物品 ID、用户 ID 和操作 ID 数据中包含任何 " 字符或特殊字符。
- 如果您的数据包含任何非 ASCII 编码字符，则必须采用 UTF-8 格式编码 CSV 文件。

- 确保按照非结构化文本元数据中所述设置所有文本数据的格式。

为训练准备物品作交互数据

物品交互是用户与您目录中的物品之间的积极交互事件。例如，用户在看电影、查看商品列表或购买一双鞋。您可以将有关用户与您的物品交互的数据导入到物品交互数据集中。您可以记录多种事件类型，例如点击、观看 或购买。

例如，如果用户单击 某件物品，然后点赞 了该物品，则可以让 Amazon Personalize 使用这些事件作为训练数据。对于每个事件，您将记录用户的 ID、物品的 ID、时间戳（采用 Unix 时间纪元格式）和事件类型（单击 和点赞）。然后，您可以将这两个物品交互事件添加到物品交互数据集中。

对于所有域应用场景和自定义配方，批量物品交互数据必须在 CSV 文件中。每一行都应代表用户和物品之间的单次交互。完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

以下各部分提供了有关如何为 Amazon Personalize 准备物品交互数据的更多信息。有关所有数据类型的批量数据格式准则，请参阅批量数据格式准则

主题

- 物品交互数据要求
- 时间戳数据
- 事件类型和事件值数据
- 上下文元数据
- 展示数据
- 交互数据示例

物品交互数据要求

以下各节列出了 Amazon Personalize 的物品交互数据要求。有关其他配额，请参阅Amazon Personalize 端点和限额。

最低训练要求

对于所有域应用场景和自定义配方，您的批量物品交互数据必须具有以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个唯一的用户 ID，每位用户至少有两次物品交互。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

要创建推荐器或自定义解决方案，您必须至少创建一个物品交互数据集。

列要求

物品交互数据必须包含以下几列。

- USER_ID - 与物品进行交互的用户的唯一标识符。每个事件都必须有一个 USER_ID。它必须是 `string`，最大长度为 256 个字符。
- ITEM_ID - 用户与之进行交互的物品的唯一标识符。每个事件都必须有一个物品 ID。它必须是 `string`，最大长度为 256 个字符。
- TIMESTAMP - 事件发生的时间（采用 Unix 纪元时间格式，以秒为单位）。每个交互都必须有一个 TIMESTAMP。有关更多信息，请参阅 时间戳数据。
- EVENT_TYPE - 物品交互事件的性质，例如点击、观看或购买。对于域推荐器，必须有一个事件类型列，并且每个交互都必须有事件类型。对于所有自定义配方，建议使用 EVENT_TYPE 列，但这是可选的。如果添加了该列，则每个事件都必须有事件类型。有关更多信息，请参阅 事件类型和事件值数据。

您可以根据应用场景和数据自由地添加其他自定义列。最大可选元数据列数为 5。这些列可以包括空值。我们建议这些列的完成度至少为 70%。

时间戳数据

时间戳数据必须采用 Unix 纪元时间格式，以秒为单位。例如，日期为 2020 年 7 月 31 日的纪元时间戳（以秒为单位）为 1596238243。要将日期转换为 Unix 纪元时间戳，请使用纪元转换器 - Unix 时间戳转换器。

Amazon Personalize 使用时间戳数据来计算时效性，并识别任何基于时间的模式。它有助于 Amazon Personalize 根据用户不断变化的偏好来更新推荐内容。

事件类型和事件值数据

物品交互数据集可以存储每个交互的事件类型和事件值数据。只有自定义资源才使用事件值数据。

事件类型数据

物品交互的事件类型提供了有关其性质和意义的上下文信息。事件类型示例可能是点击、观看 或购买。Amazon Personalize 使用事件类型数据（例如点击或购买数据）来识别用户的意图和兴趣。在物品交互数据集中，不同事件类型的最大数目加上可选元数据列的总数不能超过 10。

对于域推荐器，必须有一个事件类型列，并且每个交互都必须有事件类型。对于所有自定义配方，建议使用 EVENT_TYPE 列，但这是可选的。如果添加了该列，则每个事件都必须有事件类型。

如果您创建了自定义资源，则可以按事件类型选择用于训练的事件。如果您的数据集在 EVENT_TYPE 列中有多个事件类型，并且在配置自定义解决方案时未提供事件类型，则 Amazon Personalize 会使用所有项目交互数据进行训练，无论其类型如何，权重均相同。有关更多信息，请参阅 选择用于训练的物品交互数据。

如果您有多个事件类型，并且使用了 User-Personalization-v2 配方或 Personalized-Ranking-v2 配方，则在配置自定义解决方案时，您可以为不同的类型指定不同的权重。例如，您可以配置一个解决方案，为购买事件赋予高于点击事件的权重。有关更多信息，请参阅 使用事件配置优化解决方案。

以下应用场景具有特定的事件类型要求：

VIDEO_ON_DEMAND 域应用场景

- “因为您观看过 X”至少需要 1000 个 `Watch` 事件。
- “最受欢迎”至少需要 1000 个 `Watch` 事件。

ECOMMERCE 域应用场景

- “最常查看”至少需要 1000 个 `View` 事件。
- “最畅销”至少需要 1000 个 `Purchase` 事件。

正向和负向事件类型

Amazon Personalize 假设任何交互都是正向的。与负向事件类型（例如不喜欢）的交互不一定会阻止物品出现在用户未来的推荐中。

以下是让负向事件和用户不感兴趣内容来影响推荐的方法：

- 对于所有域应用场景和 User-Personalization 配方，Amazon Personalize 都可以使用展示数据。当某个物品出现在展示数据中而用户没有选择它时，该物品不太可能出现在推荐中。有关更多信息，请参阅 展示数据。
- 如果您使用自定义资源并导入正向和负向事件类型，则只能对正向事件类型进行训练，然后筛选出用户与之进行负向交互的物品。有关更多信息，请参阅选择用于训练的物品交互数据和筛选建议和用户细分。

事件值数据（自定义资源）

事件值数据可能是用户观看过的影片的进度百分比或用户给出的评分（满分 10 分）。如果您创建自定义解决方案，则可以根据 EVENT_TYPE 和 EVENT_VALUE 列中的数据选择用于训练的记录。对于域推荐器，Amazon Personalize 不使用事件值数据，您也无法在训练前筛选事件。

要根据类型和值选择记录，请记录事件的事件类型和事件值数据。并非所有事件都必须有事件值。您为每个事件选择的值取决于您要排除的数据以及要记录的事件类型。例如，您可以匹配用户活动，例如观看 事件类型的用户观看视频的百分比。

配置解决方案时，可以将特定值设置为从训练中排除记录的阈值。例如，如果 EVENT_TYPE 为观看的事件的 EVENT_VALUE 数据是用户观看的视频的百分比，如果您将事件值阈值设置为 0.5，将事件类型设置为观看，则 Amazon Personalize 将仅使用 EVENT_VALUE 大于或等于 0.5 的观看 交互事件来训练模型。

有关更多信息，请参阅 选择用于训练的物品交互数据。

上下文元数据

对于某些食谱和推荐器使用案例，Amazon Personalize 可以在识别底层模式时使用上下文元数据，为用户显示最相关的物品。上下文元数据是您在事件发生时在用户环境中收集的交互数据，例如其位置或设备类型。在为用户获取建议时，您也可以指定用户的上下文。

包括上下文元数据，为您的用户提供更加个性化的体验，并缩短新用户的冷启动阶段。冷启动阶段是由于缺乏历史用户数据而导致推荐的相关性较低的阶段。

例如，如果您的物品交互 CSV 文件中包括带有 `tablet` 和 phone 值的 DEVICE_TYPE 列，则 Amazon Personalize 可以了解买家在不同设备上购物的方式有何不同。在为用户获取建议时，您可以指定他们的设备，这样即使用户没有交互历史记录，建议也更具相关性。

以下内容显示了如何为物品交互 CSV 文件设定格式，将 DEVICE_TYPE 列作为上下文元数据。

```
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
.....
```

对于域数据集组，以下推荐器使用案例可以使用上下文元数据：

- 为您推荐（ECOMMERCE 域）
- 热门精选（VIDEO_ON_DEMAND 域）

对于自定义资源，使用上下文元数据的食谱包括以下内容：

- User-Personalization-v2 和 User-Personalization
- Personalized-Ranking-v2 和 Personalized-Ranking

要了解如何在获取建议时包含上下文信息，请参阅利用上下文元数据提高建议的相关性。有关展示如何使用上下文元数据的端到端示例，请参阅以下 AWS 机器学习博客文章：利用上下文信息提高 Amazon Personalize 推荐的相关性。

展示数据

展示是用户与特定物品交互（例如，单击或观看）时可见的物品列表。如果您使用提供个性化或 User-Personalization 配方的域应用场景，Amazon Personalize 可以使用展示数据来指导浏览。

在浏览中，建议包括一些通常不太可能向用户建议的物品或操作，例如新物品或操作、交互量很少的物品或操作，或者根据用户以前的行为与用户不太相关的物品或操作。物品在展示数据中出现的频率越高，Amazon Personalize 将该物品包含在浏览中的可能性就越小。

在创建推荐器或解决方案时，Amazon Personalize 始终会从训练中排除展示数据。这是因为 Amazon Personalize 不会使用展示数据训练模型。相反，在获取建议时，它会使用展示数据来指导用户浏览。

展示值最多可以包含 1000 个字符（包括竖线字符）。对于域数据集组，以下推荐器应用场景可以使用展示数据：

- 为您推荐（ECOMMERCE 域）
- 热门精选（VIDEO_ON_DEMAND 域）

有关浏览的更多信息，请参阅探索。Amazon Personalize 可以对两种类型的展示进行建模：隐式展示和显式展示。

显式展示

显式展示 是您手动记录并发送到 Amazon Personalize 的展示。使用显式展示来操作 Amazon Personalize 的结果。物品的顺序没有影响。

例如，您可能有一个购物应用程序用于提供有关鞋的建议。如果您只推荐目前有库存的鞋，则可以使用显式展示来指定这些物品。使用显式展示的建议工作流程可能如下所示：

1. 您使用 Amazon Personalize the section called “GetRecommendations” API 为其中一位用户请求建议。
2. Amazon Personalize 会为使用您的模型（解决方案版本）的用户生成建议，并在 API 响应中返回这些建议。
3. 您只向用户显示有库存的推荐的鞋。
4. 对于实时增量数据导入，当您的用户与一双鞋交互（例如单击）时，您在调用 PutEvents API 时记录该选择，并在 `impression` 参数中列出有库存的推荐物品。有关代码示例，请参阅记录包含展示数据的物品交互事件。

   要在历史物品交互数据中导入展示，您可以在 csv 文件中列出显式展示，并通过“|”字符分隔每个物品。竖线字符计入 1000 个字符的限制。有关示例，请参阅设置显式展示的格式。
5. Amazon Personalize 使用展示数据来指导浏览，即未来的建议包括交互数据较少或相关性较低的新鞋。

设置显式展示的格式

要在 CSV 文件中包括明确的展示，请添加一个 IMPRESSION 列。对于每个物品交互，请添加使用竖线“|”字符分隔的 itemID 列表。竖线字符计入展示数据 1000 个字符的限制。如果您在 PutEvents 操作中包括显式展示，则可以在字符串数组中指定物品。

以下是 CSV 文件的简短摘录，该文件在 `IMPRESSION` 列中包括显式展示。

EVENT_TYPE	展示	ITEM_ID	TIMESTAMP	USER_ID
单击	73|70|17|95|96	73	1586731606	USER_1

EVENT_TYPE	展示	ITEM_ID	TIMESTAMP	USER_ID
单击	35|82|78|57|20|63|1|90|76|75|49|71|26|24|25|6	35	1586735164	USER_2
...	...	...	...	...

应用程序显示了用户 USER_1、物品 73、70、17、95 和 96，该用户最终选择了物品 73。当您根据这些数据创建新的解决方案版本时，系统将减少向用户 USER_1 推荐物品 70、17、95 和 96 的频率。

隐式展示

隐式展示 是指您向用户显示且从 Amazon Personalize 中检索的建议。您的 CSV 文件无需包括 IMPRESSION 或 RECOMMENDATION_ID 列即可使用隐式展示。相反，您可以在 PutEvents 请求中包括 `RecommendationId`（由 GetRecommendations 和 GetPersonalizedRanking 操作返回）。Amazon Personalize 会根据您的建议数据得出隐式展示。

例如，您可能有一个应用程序用于提供流视频的建议。使用隐式展示的建议工作流程可能如下所示：

1. 您使用 Amazon Personalize the section called “GetRecommendations” API 操作为其中一位用户请求视频建议。
2. Amazon Personalize 会为使用您的模型（解决方案版本）的用户生成建议，并在 API 响应中返回这些建议（含 `recommendationId`）。
3. 您在应用程序中向用户显示视频建议。
4. 当您的用户与视频交互（例如，单击）时，在调用 PutEvents API 时记录该选择，并将 `recommendationId` 作为参数包括在内。有关代码示例，请参阅记录包含展示数据的物品交互事件。
5. Amazon Personalize 使用 `recommendationId` 从之前的视频建议中获取展示数据，然后使用展示数据来指导浏览，即未来的建议包括交互数据较少或相关性较低的新视频。

   有关使用隐式展示数据记录事件的更多信息，请参阅记录包含展示数据的物品交互事件。

交互数据示例

以下交互数据表示流视频网站上的历史用户活动。您可以使用这些数据来训练模型，以根据用户的交互数据提供电影建议。请注意，EVENT_VALUE 的某些值为空。

```
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 需要 USER_ID、ITEM_ID 和 TIMESTAMP 列。USER_ID 是应用程序用户的标识符。ITEM_ID 是影片的标识符。EVENT_TYPE 和 EVENT_VALUE 是用户交互的标识符。在示例数据中，事件是 watch 和 click 事件，值是用户观看过的视频的进度百分比。TIMESTAMP 表示购买影片时的 Unix 纪元时间。

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。示例数据的架构 JSON 文件看起来就是这个样子。

```
{
  "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"
}
```

为训练准备物品元数据

物品元数据包括您的用户与之进行交互的物品的相关数字数据和分类数据。物品元数据的例子包括创建时间戳、价格、类型、描述和供应情况。您可以将有关物品的元数据导入到 Amazon Personalize 物品数据集。

根据您的域应用场景或自定义配方，物品元数据有助于 Amazon Personalize 向用户推荐更多相关的物品、更准确地预测类似物品，或者推荐更有意义的用户细分。物品元数据还有助于 Amazon Personalize 在推荐中展示新物品。物品元数据对某些域应用场景来说是必需的，而对于所有自定义配方，物品元数据是可选的。有关更多信息，请参阅将自己的应用场景与 Amazon Personalize 资源进行匹配中的域应用场景或配方的数据要求。

在训练时，Amazon Personalize 不使用非分类字符串物品数据，例如物品标题或作者数据。但是，导入这些数据仍然可以增强推荐效果。有关更多信息，请参阅 非分类字符串数据。

Amazon Personalize 在训练期间考虑的最大物品数量取决于您的应用场景或配方。只有在训练期间考虑的物品才会出现在推荐中。

- 对于 User-Personalization-v2 或 Personalized-Ranking-v2，模型在训练期间考虑的最大物品数量为 500 万。这些物品来自物品数据集和物品交互数据集。
- 对于除 User-Personalization-v2 和 Personalized-Ranking-v2 之外的所有域应用场景和自定义配方，模型在训练和生成推荐期间考虑的最大物品数量为 75 万。

对于所有域应用场景和自定义配方，批量物品数据必须在 CSV 文件中。文件中的每一行都应代表一个唯一物品。完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

以下各部分提供了有关如何为 Amazon Personalize 准备物品元数据的更多信息。有关所有数据类型的批量数据格式准则，请参阅批量数据格式准则

主题

- 物品数据要求
- 创建时间戳数据
- 分类元数据
- 非结构化文本元数据
- 数值数据
- 非分类字符串数据
- 物品元数据示例

物品数据要求

以下是 Amazon Personalize 的物品元数据要求。

如果您不确定自己是否有足够的数据或对数据质量存有疑问，则可以将您的数据导入 Amazon Personalize 数据集，然后使用 Amazon Personalize 对其进行分析。有关更多信息，请参阅 分析 Amazon Personalize 数据集中的数据质量和数量。

- 对于所有域应用场景和自定义配方，必须有一个 ITEM_ID 列来存储每个物品的唯一标识符。每个物品都必须有一个物品 ID。它必须是 `string`，最大长度为 256 个字符。
- 对于自定义配方，您的数据必须至少包含一个分类字符串或数字元数据列。物品元数据列可以包括空值。我们建议这些列的完成度至少为 70%。
- 对于域应用场景，所需的列取决于您的域。有关更多信息，请参阅VIDEO_ON_DEMAND 域要求或ECOMMERCE 域要求。
- 最大元数据列数为 100。

VIDEO_ON_DEMAND 域要求

某些应用场景需要物品元数据（请参阅 VIDEO_ON_DEMAND 使用案例）。如果元数据是可选的，我们仍建议导入物品元数据，以便获取最相关的推荐。如果导入了物品元数据，则您的数据必须包括以下列：

- ITEM_ID

- GENRES（分类 string）
- CREATION_TIMESTAMP（采用 Unix 纪元时间格式）

下面列出了其他推荐列及其所需类型。null 类型表示该列可能有缺失值。我们建议这些列的完成度至少为 70%。包括这些列可以改善推荐。

- PRICE（浮点）
- DURATION（浮点）
- GENRE_L2（分类 string、null）
- GENRE_L3（分类 string、null）
- AVERAGE_RATING（float、null）
- PRODUCT_DESCRIPTION（文本 string、null）
- CONTENT_OWNER（分类 string、null）：拥有视频的公司。例如，值可能是 HBO、Paramount 和 NBC。
- CONTENT_CLASSIFICATION（分类 string、null）：内容的评级。例如，值可能是 G、PG、PG-13、R、NC-17 和未分级。

ECOMMERCE 域要求

对于所有 ECOMMERCE 应用场景，物品元数据都是可选的。如果您有物品数据，我们建议导入这些数据，以获得最相关的推荐。如果导入了物品元数据，则您的数据必须有以下列：

- ITEM_ID
- PRICE (float)
- CATEGORY_L1（分类 string）：有关格式化分类数据的信息，请参阅分类元数据。

下面列出了其他推荐列及其所需类型。null 类型表示该列可能有缺失值。我们建议这些列的完成度至少为 70%。包括这些列可以改善推荐。

- CATEGORY_L2（分类 string、null）
- CATEGORY_L3（分类 string、null）
- PRODUCT_DESCRIPTION（文本 string、null）
- CREATION_TIMESTAMP (float)

- AGE_GROUP（分类 `string`、`null`）：物品所针对的年龄组。值可能是新生儿、婴儿、儿童和成人。
- ADULT（分类 `string`、`null`）：物品是否仅限成人，例如酒精饮料。值可能为是或否。
- GENDER（分类 `string`、`null`）：物品所针对的性别。值可以是男性、女性和男女皆宜。

创建时间戳数据

创建时间戳数据必须采用 Unix 纪元时间格式，以秒为单位。例如，日期为 2020 年 7 月 31 日的纪元时间戳（以秒为单位）为 1596238243。要将日期转换为 Unix 纪元时间戳，请使用纪元转换器 - Unix 时间戳转换器。

Amazon Personalize 使用创建时间戳数据（采用 Unix 纪元时间格式，以秒为单位），计算物品的使用年限并相应地调整建议。

如果缺少一件或多件物品的创建时间戳数据，则 Amazon Personalize 会从交互数据（如果有）中推断出此信息，并使用物品最早交互数据的时间戳作为物品的创建时间戳。如果某个物品没有交互数据，则其创建时间戳将设置为训练集中最新交互的时间戳，且 Amazon Personalize 会将其视为新物品。

分类元数据

对于某些配方和所有域应用场景，Amazon Personalize 在识别底层模式时使用分类元数据（例如物品的类型或颜色），向用户显示最相关的物品。您可以基于自己的使用案例来定义自己的值范围。分类元数据可以是任何语言。

对于具有多个类别的物品，使用竖线“|”分隔每个值。例如，对于 GENRES 字段，您的物品数据可能是 `Action|Crime|Biopic`。如果您有多个级别的分类数据，并且某些物品在层次结构中的每个级别都有多个类别，请为每个级别使用单独列，并在每个字段名称后面附加一个级别指示符：GENRES、GENRE_L2、GENRE_L3。这样，您可以根据子类别筛选推荐，即使一个物品属于多个多级类别，也是如此（有关创建和使用筛选器的信息，请参阅筛选建议和用户细分）。例如，某个视频可能包含每个类别级别的以下数据：

- GENRES：动作|冒险
- GENRE_L2：犯罪|西方
- GENRE_L3：传记片

在此示例中，该视频位于动作 > 犯罪 > 传记片层次结构和 冒险 > 西方 > 传记片层次结构中。我们建议使用的级别最高仅到 L3，但如有必要，可以使用更多级别。

分类值最多可以有 1000 个字符。如果您的物品的分类值超过 1000 个字符，则您的数据集导入作业将失败。我们建议分类列最多包含 1000 个可能的值。导入具有更多值的分类数据会对推荐产生负面影响。以下方法有助于减少分类列的可能值数量：

- 确保值遵循一致的命名惯例并检查是否有打字错误。例如，使用“男鞋”，而不是混用“男装鞋”、“男式鞋”和“男人的鞋”。
- 合并使用略有不同的术语来指代相同基础类别的相似类别，例如“鞋子”和“运动鞋”。
- 如果您的数据采用分层结构，其中更广泛的类别（例如“鞋类”）包含更具体的子类别（例如“男鞋”、“女鞋”、“童鞋”），请为每个级别使用单独的列，并在每个字段名称后面附加一个级别指示符。例如，CATEGORY_1、CATEGORY_2 和 CATEGORY_3。这样可以减少模棱两可或重叠的类别。

对于所有食谱和域，您可以导入分类数据，并使用此类数据，根据物品的属性筛选建议。有关筛选建议的信息，请参阅筛选建议和用户细分。

非结构化文本元数据

对于某些食谱和域，Amazon Personalize 可以从非结构化文本元数据（例如产品描述、产品评论或电影提要）中提取有意义的信息。Amazon Personalize 使用非结构化文本为您的用户识别相关物品，尤其是在物品是新物品或交互数据较少的情况下。您最多可添加 1 个文本字段。在物品数据集中包含非结构化文本数据，可以提高目录中新物品的点击率和对话率。

在准备非结构化文本元数据时，请将文本用双引号括起来，并删除所有换行字符。使用 \ 字符，对数据中的任何双引号或 \ 字符进行转义处理。Amazon Personalize 会按字符限制截断文本字段。确保文本中最相关的信息位于字段的开头。

除中文和日语外，所有语言的非结构化文本值最多可以包含 2 万个字符。对于中文和日语，最多可以包含 7 千个字符。Amazon Personalize 会将超过字符限制的值截断，使其符合字符限制。

您可以采用多种语言提交非结构化文本物品，但每个物品的文本只能使用一种语言。文本可使用以下语言：

- 中文（简体）
- 中文（繁体）
- English
- 法语
- 德语
- 日语

- 葡萄牙语
- 西班牙语

数值数据

Amazon Personalize 可以使用数值物品元数据（例如价格或视频时长），为用户生成更相关的推荐。这个数值数据可以表示为整数或小数。

如果您使用 User-Personalization 或 Personalized-Ranking 自定义配方，则除了最大相关性（例如更大限度地提高收入）之外，您还可以针对物品元数据相关目标优化 Amazon Personalize 解决方案。配置解决方案时，您可以在物品数据集中选择与您的目标相关的数值元数据列。例如，您可以选择 VIDEO_LENGTH 列来最大限度地提高流时长，或者选择 PRICE 列来最大限度地提高收入。

有关更多信息，请参阅 针对其他目标优化解决方案。

非分类字符串数据

除了物品 ID 外，Amazon Personalize 在训练时不使用非分类非文本字符串数据，例如物品标题或作者数据。但是，Amazon Personalize 可以将其与以下特征结合使用。非分类值最多可以有 1000 个字符。

- Amazon Personalize 可以在建议中包含物品元数据，包括非分类字符串值。您可以使用元数据来丰富用户界面中的建议，例如将导演的姓名添加到电影建议轮播中。有关更多信息，请参阅 推荐中的物品元数据。
- 如果您使用Similar-Items，则可以生成带有主题的批量建议。生成带有主题的批量建议时，必须在批量推理作业中指定物品名称列。有关更多信息，请参阅 内容生成器中带有主题的批量建议。
- 您可以创建筛选条件，以根据非分类字符串数据在建议中包含或删除物品。有关筛选条件的更多信息，请参阅筛选建议和用户细分。

物品元数据示例

CSV 文件中的前几行影片元数据看起来可能像下面这样。

```
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."
...
...
```

ITEM_ID 列为必填列，用于存储每个物品的唯一标识符。GENRE 列存储每部电影的分类元数据，DESCRIPTION 列是非结构化文本元数据。CREATION_TIMESTAMP 列按 Unix 纪元时间格式存储每个物品的创建时间（以秒为单位）。

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。上述示例数据的架构 JSON 文件看起来就是这个样子。

```
{
  "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"
}
```

为训练准备用户元数据

您可以导入 Amazon Personalize 的用户数据包括数值数据（例如，用户年龄）和分类元数据（例如性别或会员资格）。您可以将有关用户的元数据导入到 Amazon Personalize 用户数据集。

根据您的域应用场景或自定义配方，用户元数据有助于 Amazon Personalize 向用户推荐更多相关的物品，或者推荐更有意义的用户细分。训练之后，用户元数据有助于您的模型在没有任何交互数据的情况下向用户推荐物品。有关哪些应用场景或配方使用用户元数据的更多信息，请参阅将自己的应用场景与 Amazon Personalize 资源进行匹配中域应用场景或配方的数据要求。

在训练时，Amazon Personalize 不使用非分类字符串用户数据，例如用户的名称、有关用户的关键词或标签。但是，导入这些数据仍然可以增强推荐效果。有关更多信息，请参阅 非分类字符串数据。

对于所有域应用场景和自定义配方，批量用户数据必须在 CSV 文件中。文件中的每一行都应代表一个唯一用户。完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

以下各部分提供了有关如何为 Amazon Personalize 准备用户数据的更多信息。有关所有数据类型的批量数据格式准则，请参阅批量数据格式准则

主题

- 用户数据要求
- 分类元数据
- 非分类字符串数据
- 用户元数据示例

用户数据要求

以下是 Amazon Personalize 的用户数据要求。您可以根据应用场景和数据自由地添加其他自定义列。

- 您的数据必须有一个 USER_ID 列，用于存储每个用户的唯一标识符。每个用户都必须有一个用户 ID。它必须是 `string`，最大长度为 256 个字符。

- 您的数据必须至少包含一个分类字符串或数字元数据列。对于某些用户，用户元数据列可以包括空值。我们建议这些列的完成度至少为 70%。
- 最大元数据列数为 25。

如果您不确定自己是否有足够的数据或对数据质量存有疑问，则可以将您的数据导入 Amazon Personalize 数据集，然后使用 Amazon Personalize 对其进行分析。有关更多信息，请参阅 分析 Amazon Personalize 数据集中的数据质量和数量。

分类元数据

对于一些配方以及所有域应用场景，Amazon Personalize 在识别底层模式时使用分类元数据（例如用户的性别、兴趣或会员状态），向用户显示最相关的物品。您可以基于自己的使用案例来定义自己的值范围。分类元数据可以是任何语言。

对于具有多个类别的用户，使用竖线“|”分隔每个值。例如，对于 INTERESTS 字段，您的用户数据可能是 `Movies|TV Shows|Music`。

对于所有食谱和域，您可以导入分类元数据，并使用此类数据，根据用户的属性筛选建议。有关筛选建议的信息，请参阅筛选建议和用户细分。

分类值最多可以有 1000 个字符。如果您的用户的分类值超过 1000 个字符，则您的数据集导入作业将失败。

非分类字符串数据

在训练时，Amazon Personalize 不使用除用户 ID 之外的非分类字符串用户数据，例如用户的名称、有关用户的关键词或标签。但是，Amazon Personalize 可以在筛选推荐时使用该数据。您可以创建筛选条件，以便根据您正在为其获取建议的用户（CurrentUser）的非分类字符串数据，在推荐中包括或删除物品。有关筛选条件的更多信息，请参阅筛选建议和用户细分。非分类值最多可以有 1000 个字符。

用户元数据示例

CSV 文件中的前几行用户元数据看起来可能像下面这样。

```
USER_ID,AGE,GENDER,INTEREST
5,34,Male,hiking
6,56,Female,music
8,65,Male,movies|TV shows|music
...
```

```
...
```

USER_ID 列为必填列，用于存储每个用户的唯一标识符。AGE 列是数字元数据。GENDER 和 INTEREST 列存储每个用户的分类元数据。

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。上述示例数据的架构 JSON 文件看起来就是这个样子。

```
{
  "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"
}
```

为训练准备操作元数据

操作是指您可能想向用户建议的交互或创收活动。操作可能包括安装您的移动应用、填写会员资料、加入忠诚度计划或注册促销电子邮件。您可以将有关操作的数据导入到 Amazon Personalize 操作数据集。操作数据的示例包括操作的唯一 ID、操作的估计值或操作的到期时间戳。

如果您使用 Next-Best-Action，则必须导入操作元数据。使用此配方时，Amazon Personalize 会根据您导入到操作数据集的操作预测下一个最佳操作。没有其他配方或应用场景会使用操作元数据。您不能在域数据集组中创建操作数据集。

在训练时，Amazon Personalize 不使用非分类字符串操作数据，例如操作标题或标签。但是，导入这些数据仍然可以增强推荐效果。有关更多信息，请参阅 非分类字符串数据。

批量操作数据必须保存在 CSV 文件中。文件中的每一行都应代表一个唯一操作。完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

以下各部分提供了有关如何为 Amazon Personalize 准备操作元数据的更多信息。有关所有数据类型的批量数据格式准则，请参阅批量数据格式准则

主题

- 操作数据要求
- 操作过期时间戳数据
- 重复频率数据
- Value data
- 创建时间戳数据
- 分类元数据
- 非分类字符串数据
- 操作元数据示例

操作数据要求

以下是 Amazon Personalize 的操作数据要求。

- 必须有一个 ACTION_ID 列，用于存储每个操作的唯一标识符。每个操作都必须有一个物品 ID。它必须是 `string`，最大长度为 256 个字符。
- 您的数据必须至少包含一个分类字符串或数字元数据列。操作元数据列可以包括空值。我们建议这些列的完成度至少为 70%。
- 在模型训练期间，Amazon Personalize 最多会考虑 1000 个操作。如果您导入的操作超过 1000 个，Amazon Personalize 会决定哪些操作要包含在训练中，并优先考虑新操作（您最近添加的没有交互的操作）和具有最近交互数据的现有操作。

- 最大列数为 10。

操作过期时间戳数据

操作过期时间戳指定操作不再有效的日期。您可以按 Unix 纪元时间格式提供操作过期时间戳数据（以秒为单位）。如果某操作已过期，Amazon Personalize 将不会将其包含在建议中。

如果您想将操作在建议中的出现限制在特定的时间范围内，请为操作指定操作过期时间戳。例如，您可能有一个应用程序在某个月份内开展会员资格活动。您可以为该月底的注册操作设置一个到期时间戳。到此日期后，Amazon Personalize 会自动停止建议此操作。

如果您将新操作的过期时间戳设置为过去的某个时间，或者将操作时间戳更新为过去的时间，则从建议中删除该操作最多可能需要 2 个小时。

重复频率数据

重复频率数据根据用户在您的操作交互数据集中的历史记录，指定 Amazon Personalize 在用户与特定操作交互后应等待多少天才能建议该操作。您可以指定操作的重复频率（以天为单位），最多为 30 天。

例如，您可能有一个电子商务应用程序，其中每个用户都创建一个账户和一个配置文件。如果您有一项 `complete profile` 操作，并且想在用户与该操作交互后等待一周后，然后再次建议该操作，则可以指定 7 天作为该操作的 `REPEAT_FREQENCY`。7 天后，Amazon Personalize 才开始考虑建议此操作。

如果您没有为某项操作提供重复频率，Amazon Personalize 将不会对该操作在建议中出现的次数设置任何限制。

Value data

值数据是每个操作的商业价值或重要性。操作的 `value` 可以是 1 – 10，其中 10 指数据集中最有价值的操作。

例如，您可能有两个操作，一个用于注册基本订阅，另一个用于注册高级服务。对于基本服务，您可以指定值 `5`，而对于高级服务，则指定值 `10`。

在确定向用户建议的最佳操作时，Amazon Personalize 使用值数据作为一项输入。例如，如果用户采取一项或另一项操作的可能性相同，则 Amazon Personalize 会在建议中将值最高的操作排在更高的位置。

创建时间戳数据

Amazon Personalize 使用创建时间戳数据（采用 Unix 纪元时间格式，以秒为单位），计算操作的使用年限并相应地调整建议。

如果您没有创建时间戳数据，Amazon Personalize 从任何操作交互数据中推断出这些信息。它使用操作的最旧交互数据的时间戳作为操作的创建时间戳。如果某个操作没有交互数据，则其创建时间戳将设置为训练集中最新交互的时间戳，且 Amazon Personalize 会将其视为新操作。

分类元数据

Amazon Personalize 使用有关操作的分类元数据，例如季节性或操作排他性，来确定能为用户揭示最佳操作的潜在模式。您可以基于自己的使用案例来定义自己的值范围。分类元数据可以是任何语言。

您可以导入分类数据，并使用它根据操作的属性筛选建议。有关筛选建议的信息，请参阅筛选建议和用户细分。

分类值最多可以有 1000 个字符。如果操作的分类值超过 1000 个字符，则您的数据集导入作业将失败。

非分类字符串数据

除了操作 ID 外，Amazon Personalize 在训练时不使用非分类字符串数据，例如操作的名称、有关操作的关键词或标签。但是，Amazon Personalize 可以在筛选推荐时使用该数据。您可以创建筛选条件，以根据非分类字符串数据在推荐中包括或删除操作。有关筛选条件的更多信息，请参阅筛选建议和用户细分。非分类值最多可以有 1000 个字符。

操作元数据示例

CSV 文件中的前几行操作元数据看起来可能像下面这样。

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

ACTION_ID 列是必填的。MEMBERSHIP_LEVEL 列是一个分类字符串字段。VALUE、CREATION_TIMESTAMP 和 REPEAT_FREQUENCY 字段是具有必需类型的保留关键字。

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。上述示例数据的架构 JSON 文件看起来就是这个样子。

```
{
  "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"
```

```
}
```

为训练准备操作交互数据

如果您使用 Next-Best-Action 自定义配方，Amazon Personalize 会使用操作交互数据来识别用户兴趣并预测他们最有可能采取的操作。操作交互是操作数据集中涉及用户和操作的交互。例如，如果您的操作数据集中有一个注册操作，而用户执行了此操作，则您将记录用户的 ID、操作的 ID、时间戳，并对于事件类型记录 TAKEN。

您可以将操作交互导入 Amazon Personalize 操作交互数据集中。您可以使用数据集导入作业批量导入操作交互事件，也可以通过 PutActionInteractions API 操作实时流式传输这些事件。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。

批量操作交互数据必须保存在 CSV 文件中。文件中的每一行都应代表用户和操作之间的唯一交互。完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

以下各部分提供了有关如何为 Amazon Personalize 准备操作交互数据的更多信息。有关所有数据类型的批量数据格式准则，请参阅批量数据格式准则。

主题

- 操作交互数据要求
- 事件类型数据
- 操作交互数据示例

操作交互数据要求

对操作交互数据没有最低要求。我们建议您将其导入以获取高质量的操作建议。如果您没有操作交互数据，则可以创建一个空的操作交互数据集，并使用 PutActionInteractions API 操作记录客户与操作的交互。

操作交互数据必须至少包含以下几列。您可以根据应用场景和数据自由地添加其他自定义列。

- USER_ID - 与物品进行交互的用户的唯一标识符。每个事件都必须有一个 USER_ID。它必须是 `string`，最大长度为 256 个字符。
- ACTION_ID - 用户与之进行交互的物品的唯一标识符。每个事件都必须有一个物品 ID。它必须是 `string`，最大长度为 256 个字符。

- TIMESTAMP - 事件发生的时间（采用 Unix 纪元时间格式，以秒为单位）。每个操作交互都必须有一个 TIMESTAMP。有关更多信息，请参阅 时间戳数据。
- EVENT_TYPE - 指明操作是已执行、未执行还是已查看。每个操作交互都必须有一个事件类型。有关更多信息，请参阅 事件类型数据。

在您导入操作交互数据之前，Amazon Personalize 会在您的操作数据集中推荐不进行个性化的操作，倾向分数为 0.0。在操作达到以下条件后，该操作将获得分数：

- 至少 50 次 TAKEN 事件类型的操作交互。
- 至少 50 次 NOT_TAKEN 或 VIEWED 事件类型的操作交互。

这些操作交互必须出现在最新的解决方案版本训练中，并且必须在操作交互数据集中的最新交互时间戳后的 6 周内发生。

事件类型数据

Amazon Personalize 可以使用事件类型数据中的模式来识别您的用户最有可能采取的操作。例如，如果客户经常忽略电子邮件订阅操作（以 NOT_TAKE 事件类型表示），Amazon Personalize 可能会调整建议，以减少此类操作的次数。

您只能将以下事件类型用于操作交互事件。Amazon Personalize 使用这些事件来了解您的用户，并计算接下来要建议的操作。

- 已执行 - 当用户采取了建议的操作时，记录已执行事件。
- 未执行 – 当您的用户在查看操作后特意选择不采取该操作时，记录未执行事件。例如，当您向他们展示操作时，如果他们选择否。未执行事件可能表示客户对该操作不感兴趣。
- 已查看 - 当您在用户选择采取或不采取操作之前向他们展示操作时，记录已查看事件。Amazon Personalize 使用查看事件来了解用户的兴趣。例如，如果用户查看了某项操作但未执行该操作，则该用户将来可能对此操作不感兴趣。

操作交互数据示例

包含操作交互数据和所有必填列的 CSV 文件的前几行可能看起来如下所示。

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

完成数据准备后，就可以创建架构 JSON 文件。此文件向 Amazon Personalize 提供关于数据结构的信息。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。上述示例数据的架构 JSON 文件看起来就是这个样子。

```
{

  "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"
}
```

为 Amazon Personalize 架构创建架构 JSON 文件

准备好数据后，就可以开始为要导入的每种数据类型创建架构 JSON 文件。这些文件概述了数据的结构和内容，包括列名及其数据类型。

在 创建架构和数据集 中创建 Amazon Personalize 架构时，使用架构 JSON 文件。在 Amazon Personalize 中，架构 是一种资源，它让 Amazon Personalize 可以在您将数据导入数据集时对其进行解析。您可以为要使用的每个数据集创建一个架构。

对于自定义资源，每个数据集都有特定的架构要求。对于域数据集组，您选择的域将决定您的数据集和架构要求。每个域都有每种数据集类型的默认架构。在创建数据集时，可以使用现有域架构，也可以通过修改现有默认架构来创建新的域架构。使用默认架构作为指南，确定要为域导入哪些数据。

以下各节介绍了为每种数据集类型创建架构 JSON 文件的自定义要求和域要求。

主题

- 架构格式设置要求
- VIDEO_ON_DEMAND 数据集和架构
- ECOMMERCE 数据集和架构
- 自定义数据集和架构

架构格式设置要求

为域数据集组或自定义数据集组中的数据集创建架构时，您必须遵循以下准则：

- 您必须以 Avro 格式定义架构。有关我们支持的 Avro 数据类型的信息，请参阅架构数据类型。
- 架构具有名称键，其值必须与数据集类型相匹配。
- 架构字段可以采用任何顺序显示，但它们必须与 CSV 文件中对应列标题的顺序匹配。
- 架构必须是没有嵌套结构的平面 JSON 文件。例如，一个字段不能是多个子字段的父字段。
- Amazon Personalize 架构不支持复杂类型，例如数组和映射。
- 架构字段必须具有唯一的字母数字名称。例如，您不能同时添加 `GENRES_FIELD_1` 字段和 `GENRESFIELD1` 字段。
- 您必须定义必填字段作为其所需的数据类型。保留的分类字符串字段的 `categorical` 属性必须设置为 `true`，而保留的字符串字段不能是分类字段。关键字不能出现在数据中。

- 如果您添加自己的类型为 `string` 的元数据字段，并且您希望 Amazon Personalize 在训练时使用它，则该字段必须包含 `categorical` 属性或 `textual` 属性（只有物品架构支持带有文本属性的字段）。
- Amazon Personalize 在训练或筛选建议时不使用 `boolean` 类型数据。要让 Amazon Personalize 在训练或筛选时使用布尔数据，请使用类型为字符串 的字段，并在数据中使用值 `"True"` 和 `"False"`。或者，您可以使用类型 int 或 long 以及值 `0` 和 `1`。
- 文本字段的类型必须为 `string`，且必须将 `textual` 属性设置为 `true`。有关非结构化文本数据的更多信息，请参阅非结构化文本元数据。

根据域和数据集类型，域数据集组数据集还有其他要求。自定义数据集组数据集也有其他要求，具体取决于类型。

架构数据类型

Amazon Personalize 架构支持以下 Avro 类型的字段：

- 浮点数
- double
- 整数
- 长整数
- 字符串
- 布尔值
- null

一些必填字段和保留字段支持空数据。通过向字段添加 `null` 类型，您可以使用不完美的数据（例如，空值的元数据）来生成建议。有关哪些字段支持空数据的信息，请参阅域的架构要求主题：VIDEO_ON_DEMAND 数据集和架构、ECOMMERCE 数据集和架构 或 自定义数据集和架构。以下示例说明了如何为 GENDER 字段添加空类型。

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
```

```
}
```

VIDEO_ON_DEMAND 数据集和架构

当您为 VIDEO_ON_DEMAND 域创建域数据集组时，每种数据集类型都有一个默认架构，其中包含一组 VIDEO_ON_DEMAND 特定的必填字段和建议字段。您可以使用默认架构，也可以基于默认架构创建新架构。您导入的数据在格式和类型上必须与您的架构相匹配。使用以下各部分中列出的默认域架构作为指南，确定要导入哪些数据来创建基于 VIDEO_ON_DEMAND 的推荐器。

您可以自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

以下主题提供了有关 VIDEO_ON_DEMAND 域中每个数据集的必填字段和建议字段的信息。每个数据集部分都包含 JSON 格式的默认 VIDEO_ON_DEMAND 架构。

主题

- VIDEO_ON_DEMAND 域数据集和架构要求
- 物品交互数据集要求（VIDEO_ON_DEMAND 域）
- 用户数据集要求（VIDEO_ON_DEMAND 域）
- 物品数据集要求（VIDEO_ON_DEMAND 域）

VIDEO_ON_DEMAND 域数据集和架构要求

每个数据集类型都具有以下必填字段和保留关键字。保留关键字是可选的非元数据字段。这些字段被视为保留字段，因为您在使用这些字段时必须将它们定义为所需的数据类型。保留的分类字符串字段的 `categorical` 必须设置为 `true`，而保留的字符串字段不能是分类字段。关键字不能出现在数据中。

数据集类型	必填字段	保留关键字
物品交互（默认架构）	USER_ID (`string`) ITEM_ID (`string`) TIMESTAMP (`long`)	EVENT_VAL UE（`float`、`null`） IMPRESSIO N（`string`、`null`）

数据集类型	必填字段	保留关键字
	EVENT_TYPE（string，取决于使用案例，Watch 和 Click 事件类型）	RECOMMENDATION_ID（string、null） EVENT_ATTRIBUTION_SOURCE（string、null）
用户（默认架构）	USER_ID (string) 1 个元数据字段（分类 string 或数字）	SUBSCRIPTION_MODEL（分类 string、null）
物品（默认架构）	ITEM_ID (string) CREATION_TIMESTAMP (long) GENRES（分类 string）	PRICE（float、null） DURATION（float、null） GENRE_L2（分类 string、null） GENRE_L3（分类 string、null） AVERAGE_RATING（float、null） PRODUCT_DESCRIPTION（文本 string、null） CONTENT_OWNER（分类 string、null） CONTENT_CLASSIFICATION（分类 string、null）

物品交互数据集要求（VIDEO_ON_DEMAND 域）

物品交互数据集存储来自用户和 VIDEO_ON_DEMAND 目录中物品之间交互的历史数据和实时数据。有关您可以存储在交互数据集中的数据类型的更多信息，请参阅物品交互数据。

您必须拥有适用于所有使用案例的物品交互数据集，且您的架构必须包含以下字段：

- USER_ID (`string`)
- ITEM_ID `string`
- TIMESTAMP (`long`)
- EVENT_TYPE（`string`，取决于使用案例，`Watch` 和 `Click` 事件类型）

您的架构还可以包含以下保留关键字：

- EVENT_VALUE（`float`、`null`）
- IMPRESSION（`string`、`null`）
- RECOMMENDATION_ID（`string`、`null`）

您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。有关 VIDEO_ON_DEMAND 域的物品交互数据集的默认架构示例，请参阅默认交互架构（VIDEO_ON_DEMAND 域）。

如果您的事件具有值数据（例如观看视频的百分比），则可以选择添加保留关键字 EVENT_VALUE。如果您想包含显式和隐式展示数据，则可以选择添加保留关键字 IMPRESSION。有关记录展示数据的更多信息，请参阅展示数据。

您可以添加到物品交互数据集的最大可选元数据字段总数，再加上物品交互数据中不同事件类型的总数，结果为 10。此计数中包含的元数据字段是 EVENT_TYPE、EVENT_VALUE 字段以及您添加到架构中的任何自定义元数据字段。不包括保留字段（例如 IMPRESSION）的元数据字段的最大数量为 5。分类值最多可以有 1000 个字符。如果交互的分类值大于 1000，则您的数据集导入作业将失败。

有关 VIDEO_ON_DEMAND 域的物品交互数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

默认交互架构（VIDEO_ON_DEMAND 域）

以下是物品交互数据集的默认 VIDEO_ON_DEMAND 域架构。

```
{
```

```
  "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"
}
```

用户数据集要求（VIDEO_ON_DEMAND 域）

用户数据集 存储有关用户的元数据。这可能包括每个用户的年龄、兴趣、性别或会员资格等信息。有关您可以导入 Amazon Personalize 的用户数据类型的信息，请参阅用户元数据。有关通用 Amazon Personalize 架构要求的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

对于所有 VIDEO_ON_DEMAND 使用案例，用户数据集都是可选的。如果您有用户数据，我们建议您创建一个用户数据集，以获得最相关的建议。如果您创建用户数据集，则架构必须包含以下字段。

- USER_ID
- 1 个元数据字段（分类 `string` 或数字）

您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。有关 VIDEO_ON_DEMAND 域用户数据集的默认架构示例，请参阅默认用户架构（VIDEO_ON_DEMAND 域）。

默认架构中包含 SUBSCRIPTION_MODEL 字段。此字段是一个可选保留关键字字段，其类型必须为 string，类别设置为 true。为了获得最佳建议，我们建议在您的数据中存在关于每个用户的订阅模式信息时，在架构中保留此字段。您导入的数据必须与您的架构相匹配。

使用分类数据

要使用分类数据，请添加类型为 string 的字段，并在架构中将该字段的类别属性设置为 true。然后，将分类数据包含在批量 CSV 文件和单个记录导入中。对于具有多个类别的用户，使用竖线“|”分隔每个值。例如，对于 SUBSCRIPTION_MODEL 字段，您的用户数据可能是 student|monthly|discount。

分类值最多可以有 1000 个字符。如果您的用户的分类值超过 1000 个字符，则您的数据集导入作业将失败。

默认用户架构（VIDEO_ON_DEMAND 域）

以下是用户数据集的默认 VIDEO_ON_DEMAND 域架构。

```
{

  "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"
}
```

物品数据集要求（VIDEO_ON_DEMAND 域）

物品数据集 存储有关目录物品的元数据。这可能包括每件物品的价格、类型和库存等信息。有关您可以导入 Amazon Personalize 的物品数据类型的信息，请参阅物品元数据。有关 Amazon Personalize

架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

某些使用案例需要物品数据集（请参阅 VIDEO_ON_DEMAND 使用案例）。可选时，我们仍建议您创建一个物品数据集，以获取最相关的建议。如果您创建物品数据集，则架构必须包含以下字段：

- ITEM_ID
- GENRES（分类 `string`）
- CREATION_TIMESTAMP（采用 Unix 纪元时间格式）

您的架构还可以包含以下保留关键字。每个关键字都列出了其所需的数据类型以及它是否支持空数据。添加空类型是可选的。

- PRICE（浮点）
- DURATION（浮点）
- GENRE_L2（分类 `string`、`null`）
- GENRE_L3（分类 `string`、`null`）
- AVERAGE_RATING（`float`、`null`）
- PRODUCT_DESCRIPTION（文本 `string`、`null`）
- CONTENT_OWNER（分类 `string`、`null`）：拥有视频的公司。例如，值可能是 HBO、Paramount 和 NBC。
- CONTENT_CLASSIFICATION（分类 `string`、`null`）：内容的评级。例如，值可能是 G、PG、PG-13、R、NC-17 和未分级。

为了获得最佳建议，我们建议您在架构中尽可能多保留这些字段，因为您拥有数据。您导入的数据必须与您的架构相匹配。最大元数据列数为 100。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

对于具有多个多级类别的物品，使用保留关键字 GENRE_L2 和 GENRE_L3。有关更多信息，请参阅使用分类数据。有关文本和分类元数据的信息，请参阅为训练准备物品元数据。有关 ECOMMERCE 域物品数据集的默认架构示例，请参阅默认物品架构（VIDEO_ON_DEMAND 域）。

使用分类数据

要使用分类数据，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和单个物品导入中。分类值最多可以有 1000 个字符。如果您的物品的分类值超过 1000 个字符，则您的数据集导入作业将失败。

对于具有多个类别的物品，使用竖线“|”分隔每个值。例如，对于 GENRES 字段，您的物品数据可能是 `Action|Crime|Biopic`。如果您有多个级别的分类数据，并且某些物品在层次结构中的每个级别都有多个类别，请为每个级别添加一个字段，并在每个字段名称后面附加一个级别指示符：GENRES、GENRE_L2、GENRE_L3。这样，您可以根据子类别筛选建议，即使一个物品属于多个多级类别，也是如此。例如，某个视频可能包含每个类别级别的以下数据：

- GENRES：动作|冒险
- GENRE_L2：犯罪|西方
- GENRE_L3：传记片

在此示例中，该视频位于动作 > 犯罪 > 传记片层次结构和 冒险 > 西方 > 传记片层次结构中。我们建议使用的级别最高仅到 L3，但如有必要，可以使用更多级别。有关创建和使用筛选器的信息，请参阅筛选建议和用户细分。

默认物品架构（VIDEO_ON_DEMAND 域）

以下是 VIDEO_ON_DEMAND 域物品数据集的默认架构。

```
{
  "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"
}
```

ECOMMERCE 数据集和架构

当您为 ECOMMERCE 域创建域数据集组时，每种数据集类型都有一个默认架构，其中包含一组 ECOMMERCE 特定的必填字段和建议字段。您可以使用默认架构，也可以基于默认架构创建新架构。您导入的数据在格式和类型上必须与您的架构相匹配。使用以下各部分中列出的默认域架构作为指南，确定要导入哪些数据来创建基于 ECOMMERCE 的推荐器。

您可以自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

以下主题提供了有关 ECOMMERCE 域中每个数据集的必填字段和建议字段的信息。每个数据集部分都包含 JSON 格式的默认 ECOMMERCE 架构。

主题

- ECOMMERCE 域数据集和架构要求
- 物品交互数据集要求（ECOMMERCE 域）
- 用户数据集要求（ECOMMERCE 域）
- 物品数据集要求（ECOMMERCE 域）

ECOMMERCE 域数据集和架构要求

每个数据集类型都具有以下必填字段和保留关键字。保留关键字是可选的非元数据字段。这些字段被视为保留字段，因为您在使用这些字段时必须将它们定义为所需的数据类型。保留的分类字符串字段的 `categorical` 必须设置为 `true`，而保留的字符串字段不能是分类字段。关键字不能出现在数据中。

数据集类型	必填字段	保留关键字
物品交互（默认架构）	USER_ID (string) ITEM_ID (string) TIMESTAMP (long) EVENT_TYPE（string，取决于使用案例，Purchase 和 View 事件类型）	EVENT_VALUE（float、null） IMPRESSION（string、null） RECOMMENDATION_ID（string、null） EVENT_ATTRIBUTION_SOURCE（string、null）
用户（默认架构）	USER_ID (string) 1 个元数据字段（分类 string 或数字）	
物品（默认架构）	ITEM_ID (string) PRICE (float) CATEGORY_L1（分类 string）	CATEGORY_L2（分类 string、null） CATEGORY_L3（分类 string、null） PRODUCT_DESCRIPTION（文本 string、null） CREATION_TIMESTAMP (long) AGE_GROUP（分类 string、null） ADULT（分类 string、null） GENDER（分类 string、null）

物品交互数据集要求（ECOMMERCE 域）

物品交互数据集存储来自用户和 ECOMMERCE 目录中物品之间交互的历史数据和实时数据。有关您可以存储在交互数据集中的数据类型的更多信息，请参阅物品交互数据。有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

您必须至少创建一个物品交互数据集，且您的架构必须包含以下字段：

- USER_ID (`string`)
- ITEM_ID (`string`)
- TIMESTAMP (`long`)
- EVENT_TYPE（`string`，取决于使用案例，`Purchase` 和 `View` 事件类型）

您的架构还可以包含以下保留关键字：

- EVENT_VALUE（`float`、`null`）
- IMPRESSION（`string`、`null`）
- RECOMMENDATION_ID（`string`、`null`）

您导入的数据必须与您的架构相匹配。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。有关 ECOMMERCE 域的物品交互数据集的默认架构示例，请参阅默认交互架构（ECOMMERCE 域）。

如果您的事件具有值数据，则可以选择添加保留关键字 EVENT_VALUE。如果您想包含显式和隐式展示数据，则可以选择添加保留关键字 IMPRESSION。有关记录展示数据的更多信息，请参阅展示数据。

您可以添加到物品交互数据集的最大可选元数据字段总数，再加上物品交互数据中不同事件类型的总数，结果为 10。此计数中包含的元数据字段是 EVENT_TYPE、EVENT_VALUE 字段以及您添加到架构中的任何自定义元数据字段。不包括保留字段（例如 IMPRESSION）的元数据字段的最大数量为 5。分类值最多可以有 1000 个字符。如果交互的分类值大于 1000，则您的数据集导入作业将失败。

有关 ECOMMERCE 域的物品交互数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

默认交互架构（ECOMMERCE 域）

以下是物品交互数据集的默认 ECOMMERCE 域架构。

```
{

  "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"
}
```

用户数据集要求（ECOMMERCE 域）

用户数据集 存储有关用户的元数据。这可能包括每个用户的年龄、性别或会员资格等信息。有关您可以导入 Amazon Personalize 的用户数据类型的更多信息，请参阅用户元数据。有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

对于所有 ECOMMERCE 使用案例，用户数据集都是可选的。如果您有用户数据，我们建议您创建一个用户数据集，以获得最相关的建议。如果您创建用户数据集，则架构必须包含以下字段。

- USER_ID
- 1 个元数据字段（分类 `string` 或数字）

您导入的数据必须与您的架构相匹配。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。有关 ECOMMERCE 域用户数据集的默认架构示例，请参阅默认用户架构（ECOMMERCE 域）。

有关用户数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

使用分类数据

要使用分类数据，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和单个记录导入中。对于具有多个类别的用户，使用竖线“|”分隔每个值。例如，对于 SUBSCRIPTION_MODEL 字段，您的用户数据可能是 student|monthly|discount。

分类值最多可以有 1000 个字符。如果您的用户的分类值超过 1000 个字符，则您的数据集导入作业将失败。

默认用户架构（ECOMMERCE 域）

以下是将 CATEGORY 字段作为必填元数据字段的用户数据集的默认 ECOMMERCE 域架构。

```
{

  "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"
}
```

物品数据集要求（ECOMMERCE 域）

物品数据集 存储有关 ECOMMERCE 物品的元数据。这可能包括每件物品的价格、类别和产品描述等信息。有关您可以导入 Amazon Personalize 的物品数据类型的更多信息，请参阅物品元数据。有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有架构，不论域如何。

对于所有 ECOMMERCE 使用案例，物品数据集都是可选的。如果您有物品数据，我们建议您创建一个物品数据集，以获得最相关的建议。如果您创建物品数据集，则架构必须包含以下字段：

- ITEM_ID
- PRICE (`float`)
- CATEGORY_L1（分类 `string`）

您的架构还可以包含以下保留关键字。对于分类字段，您可以根据自己的使用案例定义自己的值范围。

- CATEGORY_L2（分类 `string`、`null`）
- CATEGORY_L3（分类 `string`、`null`）
- PRODUCT_DESCRIPTION（文本 `string`、`null`）
- CREATION_TIMESTAMP (`float`)
- AGE_GROUP（分类 `string`、`null`）：物品所针对的年龄组。值可能是新生儿、婴儿、儿童和成人。
- ADULT（分类 `string`、`null`）：物品是否仅限成人，例如酒精饮料。值可能为是或否。
- GENDER（分类 `string`、`null`）：物品所针对的性别。值可以是男性、女性和男女皆宜。

为了获得最佳建议，我们建议您在架构中尽可能多保留这些字段，因为您拥有数据。您导入的数据必须与您的架构相匹配。您导入的数据必须与您的架构相匹配。最大元数据列数为 100。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

对于具有多个多级类别的物品，使用保留关键字 CATEGORY_L2 和 CATEGORY_L3。有关更多信息，请参阅 使用分类数据。有关文本和分类元数据的信息，请参阅非结构化文本元数据。有关 ECOMMERCE 域物品数据集的默认架构示例，请参阅默认物品架构（ECOMMERCE 域）。

使用分类数据

要使用分类数据，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和单个物品导入中。您可以基于自己的使用案例来定义自己的值范围。分类值最多可以有 1000 个字符。如果您的物品的分类值超过 1000 个字符，则您的数据集导入作业将失败。

对于具有多个类别的物品，使用竖线“|”分隔每个值。例如，对于 CATEGORY_L1 字段，您的物品数据可能是 `Electronics|Productivity|Mouse`。如果您有多个级别的分类数据，并且某些物品在层次结构中的每个级别都有多个类别，请为每个级别添加一个字段，并在每个字段名称后面附加一个级别指示符：CATEGORY_L1、CATEGORY_L2、CATEGORY_L3。这样，您可以根据子类别筛选建议，即使一个物品属于多个多级类别，也是如此。例如，某个物品可能包含每个类别级别的以下数据：

- CATEGORY_L1：电子产品|生产力
- CATEGORY_L2：生产力|计算机
- CATEGORY_L3：鼠标

在此示例中，该物品位于电子产品 > 生产力 > 鼠标层次结构和 生产力 > 计算机 > 鼠标层次结构中。我们建议使用的级别最高仅到 L3，但如有必要，可以使用更多级别。有关创建和使用筛选器的信息，请参阅筛选建议和用户细分。

默认物品架构（ECOMMERCE 域）

以下是仅包含必填字段的 ECOMMERCE 域的物品数据集的默认架构。

```
{
  "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"
}
```

自定义数据集和架构

创建自定义数据集组时，您可以从头开始创建自己的架构。自定义数据集组数据集和架构的必填字段更少，灵活性更高。以下主题说明了自定义数据集组的数据集的架构与数据要求。每个数据集部分都列出了数据集类型所需的数据，并提供了架构的 JSON 示例。

有关您可以导入 Amazon Personalize 的数据类型的信息，请参阅 Amazon Personalize 可以使用的数据类型。有关 Amazon Personalize 架构的一般要求（例如格式设置要求和可用的字段数据类型）的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。这些要求适用于所有 Amazon Personalize 架构。

主题

- 自定义数据集和架构要求
- 物品交互数据集架构要求（自定义）
- 用户数据集架构要求（自定义）
- 物品数据集架构要求（自定义）
- 操作数据集架构要求（自定义）
- 操作交互数据集架构要求（自定义）

自定义数据集和架构要求

在为自定义数据集组创建数据集时，每种数据集类型都具有以下必填字段和带有所需数据类型的保留关键字。

数据集类型	必填字段	保留关键字
物品交互（架构示例）	USER_ID (string)	EVENT_TYPE (string)

数据集类型	必填字段	保留关键字
	ITEM_ID (string) TIMESTAMP (long)	EVENT_VALUE (float、null) IMPRESSION (string、null) RECOMMENDATION_ID (string、null) EVENT_ATTRIBUTION_SOURCE (string、null)
用户 (架构示例)	USER_ID (string) 1 个元数据字段 (分类 string 或数字)	
物品 (架构示例)	ITEM_ID (string) 1 个元数据字段 (分类或文本 string 字段或数字字段)	CREATION_TIMESTAMP (long)
操作 (架构示例)	ACTION_ID (string) 1 个元数据字段 (分类 string 或数字)	CREATION_TIMESTAMP (long) VALUE (long、null) TYPE (string、null) EXPIRATION_TIMESTAMP (long、null) REPEAT_FREQUENCY (long、null)

数据集类型	必填字段	保留关键字
操作交互（架构示例）	USER_ID (string) ACTION_ID (string) EVENT_TYPE (string) TIMESTAMP (long)	IMPRESSION (string、null) RECOMMENDATION_ID (string、null)

元数据字段

元数据包括不需要或不使用保留关键字的字符串或非字符串字段。元数据架构具有以下限制：

- 用户、物品和操作架构至少需要一个元数据字段。
- 您最多可以为用户架构添加 25 个元数据字段，为物品架构最多添加 100 个元数据字段，并为操作架构最多添加 10 个元数据字段。
- 如果您添加自己的类型为 `string` 的元数据字段，则该字段必须包含 `categorical` 属性或 `textual` 属性（只有物品架构支持带有文本属性的字段）。否则，Amazon Personalize 在训练模型时不会使用该字段。

保留关键字

保留关键字是可选的非元数据字段。这些字段被视为保留字段，因为您在使用这些字段时必须将其定义为所需的数据类型，而且关键字不能用作数据中的值。保留的分类字符串字段的 `categorical` 必须设置为 `true`，而保留的字符串字段不能是分类字段。以下是保留关键字：

- EVENT_TYPE：对于具有一种或多种事件类型（例如单击 和下载）的物品交互数据集，使用 EVENT_TYPE 字段。您必须将 EVENT_TYPE 字段定义为 `string`，且不能将其设置为分类。
- EVENT_VALUE：对于包含事件值数据（例如用户观看视频的百分比）的物品交互数据集，使用类型为 `float` 和可选类型为 `null` 的 EVENT_VALUE 字段。
- CREATION_TIMESTAMP：对于具有每个物品创建日期时间戳的物品或操作数据集，使用类型为 `long` 的 CREATION_TIMESTAMP 字段。Amazon Personalize 使用 CREATION_TIMESTAMP 数据，计算物品的使用年限并相应地调整建议。请参阅创建时间戳数据。
- IMPRESSION：对于包含显式展示数据的物品交互数据集，使用类型为 String 和可选类型为 `null` 的 IMPRESSION 字段。展示是用户与特定物品交互（例如，单击或观看）时可见的物品列表。有关更多信息，请参阅 展示数据。

- RECOMMENDATION_ID：对于使用先前建议作为隐式展示数据的物品交互数据集，可以选择使用类型为 `String` 和可选类型为 `null` 的 `RECOMMENDATION_ID` 字段。

  在生成建议时，您无需为 Amazon Personalize 添加 `RECOMMENDATION_ID` 字段，即可使用隐式展示。您可以在没有它的情况下，在 PutEvents 操作中传递 `recommendationId`。有关更多信息，请参阅 展示数据。
- VALUE：对于操作数据集，如果您对于部分或全部操作具有数据值，请在架构中添加一个 `VALUE` 字段。对于其类型，请使用 `long` 和可选类型 `null`。有关操作及其值的更多信息，请参阅Value data。
- ACTION_EXPIRATION_TIMESTAMP：对于操作数据集，如果您的部分或全部操作有过期时间戳，请在架构中添加一个 `ACTION_EXPIRATION_TIMESTAMP` 字段。对于其类型，请使用 `long` 和可选类型 `null`。有关过期时间戳的更多信息，请参阅操作过期时间戳数据。
- REPEAT_FREQUENCY：对于操作数据集，如果您对于部分或全部操作具有重复频率数据，请在架构中添加一个 `REPEAT_FREQUENCY` 字段。对于其类型，请使用 `long` 和可选类型 `null`。有关重复频率数据的更多信息，请参阅重复频率数据。

物品交互数据集架构要求（自定义）

物品交互数据集存储来自用户和目录物品之间交互的历史数据和实时数据。有关 Amazon Personalize 可以使用的交互数据类型的信息，请参阅物品交互数据。

您为每次交互提供的数据必须与您的架构相匹配。根据您的架构，交互元数据可能包含空值。您必须为每次交互至少提供以下内容：

- 用户 ID
- 物品 ID
- 时间戳（采用 Unix 纪元时间格式）

您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

您可以添加到物品交互数据集的最大可选元数据字段总数，再加上物品交互数据中不同事件类型的总数，结果为 10。此计数中包含的元数据字段是 EVENT_TYPE、EVENT_VALUE 字段以及您添加到架构中的任何自定义元数据字段。不包括保留字段（例如 IMPRESSION）的元数据字段的最大数量为 5。分类值最多可以有 1000 个字符。如果交互的分类值大于 1000，则您的数据集导入作业将失败。

有关物品交互数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

交互架构示例（自定义）

以下示例显示了物品交互数据集的架构。USER_ID、ITEM_ID 和 TIMESTAMP 为必需字段。EVENT_TYPE、EVENT_VALUE 和 IMPRESSION 字段是 Amazon Personalize 可识别的可选保留关键字。EVENT_TYPE 必须是字符串类型，不能是分类类型。LOCATION 和 DEVICE 是可选的上下文元数据字段。有关架构要求的信息，请参阅自定义数据集和架构要求。

```
{

  "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"
}
```

对于此架构，CSV 文件中的前几行历史数据可能如下所示。请注意，EVENT_VALUE 的某些值为空。

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

用户数据集架构要求（自定义）

用户数据集 存储有关用户的元数据。这可能包括每个物品的年龄、性别或会员资格等信息。有关您可以导入 Amazon Personalize 的用户数据类型的信息，请参阅用户元数据。

您为每个用户提供的数据必须与您的架构相匹配。您必须至少为每个用户提供一个用户 ID（最大长度为 256 个字符）。根据您的架构，用户元数据可以包含 empty/null 值。您的用户架构必须至少有一个元数据字段，但是如果您添加 null 类型，则用户的该值可以为空。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

要使用分类数据，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和单个记录导入中。对于具有多个类别的用户，使用竖线“|”分隔每个值。例如，对于 SUBSCRIPTION_MODEL 字段，您的用户数据可能是 student|monthly|discount。

分类值最多可以有 1000 个字符。如果您的用户的分类值超过 1000 个字符，则您的数据集导入作业将失败。

有关用户数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

用户架构示例（自定义）

以下示例展示了如何构建用户架构。USER_ID 字段为必填字段，AGE 和 GENDER 字段为元数据字段。至少需要一个元数据字段，您最多可以添加 25 个元数据字段。有关架构要求的信息，请参阅自定义数据集和架构要求。

```
{
  "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"
}
```

对于此架构，CSV 文件中的前几行历史数据可能如下所示。

```
USER_ID,AGE,GENDER
```

```
5,34,Male
6,56,Female
8,65,Male
...
...
```

物品数据集架构要求（自定义）

物品数据集 存储有关目录物品的元数据。这可能包括每件物品的价格、类型和库存等信息。有关您可以导入 Amazon Personalize 的物品数据类型的信息，请参阅物品元数据。

您为每个物品提供的数据必须与您的物品数据集架构相匹配。您必须至少为每个物品提供一个物品 ID（最大长度为 256 个字符）。根据您的架构，项目元数据可以包含 empty/null 值。您的架构必须至少有一个元数据字段，但如果添加 `null` 类型，则物品的该值可以为空。您可以根据您的使用案例和数据自由添加其他字段。只要字段未列为必填字段或保留字段，并且数据类型列在架构数据类型中，则字段名称和数据类型由您决定。

要使用分类数据，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和单个物品导入中。分类值最多可以有 1000 个字符。如果您的物品的分类值超过 1000 个字符，则您的数据集导入作业将失败。

对于具有多个类别的物品，使用竖线“|”分隔每个值。例如，对于 GENRES 字段，您的物品数据可能是 `Action|Crime|Biopic`。如果您有多个级别的分类数据，并且某些物品在层次结构中的每个级别都有多个类别，请为每个级别添加一个字段，并在每个字段名称后面附加一个级别指示符：GENRES、GENRE_L2、GENRE_L3。这样，您可以根据子类别筛选建议，即使一个物品属于多个多级类别，也是如此（有关创建和使用筛选器的信息，请参阅筛选建议和用户细分）。例如，某个视频可能包含每个类别级别的以下数据：

- GENRES：动作|冒险
- GENRE_L2：犯罪|西方
- GENRE_L3：传记片

在此示例中，该视频位于动作 > 犯罪 > 传记片层次结构和 冒险 > 西方 > 传记片层次结构中。我们建议使用的级别最高仅到 L3，但如有必要，可以使用更多级别。

在模型训练期间，Amazon Personalize 最多会考虑 75 万件物品。如果您导入的物品超过 75 万件，则 Amazon Personalize 会决定在训练中包含哪些物品，重点是包括新物品（您最近添加的没有交互的物品）以及含最近交互数据的现有物品。

有关物品数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

物品数据集架构示例（自定义）

以下示例展示了如何构建物品架构。ITEM_ID 字段为必填项。GENRE 字段是分类元数据字段，DESCRIPTION 字段是文本元数据字段。至少需要一个元数据字段。您最多可以添加 100 个元数据字段。CREATION_TIMESTAMP 字段是保留关键字字段。有关架构要求的信息，请参阅自定义数据集和架构要求。

```
{
  "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"
}
```

对于此架构，CSV 文件中的前几行历史数据可能如下所示。

```
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."
...
...
```

操作数据集架构要求（自定义）

操作是您可能需要向客户建议的交互活动。操作可能包括安装您的移动应用、填写会员资料、加入忠诚度计划或注册促销电子邮件。操作数据集存储有关您的操作的数据。有关您可以导入 Amazon Personalize 的操作数据类型的信息，请参阅操作元数据。

您为每个操作提供的数据必须与您的操作数据集架构相匹配。根据您的架构，操作元数据可以包含 empty/null 值。

您必须至少为每个物品提供一个操作 ID（最大长度为 256 个字符）。您的架构必须至少有一个元数据字段，但如果您添加 `null` 类型，则操作的这个值可以为空。您可以根据您的使用案例和数据添加其它字段。您可以选择字段名称和数据类型，除非字段列出为必需或保留，并且数据类型在架构数据类型中列出。

要添加分类字段，请添加类型为 `string` 的字段，并在架构中将该字段的类别属性设置为 `true`。然后，将分类数据包含在批量 CSV 文件和各个操作导入中。分类值最多可以有 1000 个字符。如果操作的分类值超过 1000 个字符，则您的数据集导入作业将失败。

有关操作数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

操作数据集架构示例（自定义）

以下示例展示了如何构建操作架构。`ACTION_ID` 字段为必填项。`MEMBERSHIP_LEVEL` 字段是一个分类字符串字段。`VALUE`、`CREATION_TIMESTAMP` 和 `REPEAT_FREQUENCY` 字段是具有必需类型的保留关键字。您最多可添加 10 列。有关架构要求的信息，请参阅自定义数据集和架构要求。

```
{
```

```
  "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"
}
```

对于此架构，CSV 文件中的前几行历史数据可能如下所示。

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

操作交互数据集架构要求（自定义）

操作交互数据集存储来自用户与操作数据集中的操作之间交互的历史数据和实时数据。有关 Amazon Personalize 可以使用的数据类型的信息，请参阅操作交互数据。

您为每次交互提供的数据必须与您的架构相匹配。根据您的架构，交互元数据可能包含空值。您的架构必须至少包含以下内容：

- USER_ID
- ACTION_ID
- TIMESTAMP
- EVENT_TYPE

您可以根据您的使用案例和数据添加其它字段。您可以选择字段名称和数据类型，除非字段列出为必需或保留，并且数据类型在架构数据类型中列出。

有关操作交互数据集的最低要求和最大数据限制的更多信息，请参阅服务配额。

操作交互数据集架构示例（自定义）

以下示例显示了只包含必需字段的操作交互数据集的架构。有关一般架构格式要求的信息，请参阅架构格式设置要求。

```
{

  "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"
}
```

对于此架构，CSV 文件中的前几行历史数据可能如下所示。

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

准备清单

此清单列出了 Amazon Personalize 的特征、要求和数据指南。它可以帮助您进行规划，也可以将其用作在 Amazon Personalize 中创建资源时的参考。

主题

- 您是否将自己的使用案例与 Amazon Personalize 资源进行了匹配？
- 您是否有足够的物品交互数据？
- 您是否有实时事件流式传输架构？
- 您的数据是否针对 Amazon Personalize 进行了优化？
- 您是否收集可以改进建议的可选数据？
- 您是否有计划测试您的建议？
- 您是否还有其他业务目标？

您是否将自己的使用案例与 Amazon Personalize 资源进行了匹配？

Amazon Personalize 建议可以解决以下使用案例：

- 为用户生成个性化建议
- 推荐类似或相关物品
- 推荐趋势或热门物品
- 为用户建议下一个最佳操作
- 按相关性重新排序（仅限使用自定义资源）
- 生成用户细分（仅限使用自定义资源）

Amazon Personalize 提供基于域的资源和为这些使用案例配置的自定义资源。首先，创建域数据集组或自定义数据集组：

- 对于域数据集组，创建针对 VIDEO_ON_DEMAND 或 ECOMMERCE 域进行预配置和优化的资源。

  如果您有流视频或电子商务应用程序，我们建议您从域数据集组开始。您仍然可以添加自定义资源，例如针对自定义使用案例训练的解决方案和解决方案版本。而且，您仍可以使用自定义资源来获取批量推荐。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。

- 对于自定义数据集组，选择与您的使用案例相匹配的食谱。然后，只训练和部署可配置的解决方案与解决方案版本（经过训练的 Amazon Personalize 建议模型）。准备好之后，您可以在市场活动中部署解决方案版本来进行实时推荐。或者，您也可以在没有市场活动的情况下获取批量推荐。

  如果您没有流视频或电子商务应用程序，我们建议您创建自定义数据集组。否则，从域数据集组开始，然后根据需要添加自定义资源。

有关 Amazon Personalize 中提供的使用案例和自定义食谱的信息，请参阅将自己的应用场景与 Amazon Personalize 资源进行匹配。

您是否有足够的物品交互数据？

对于所有使用案例和配方，您的 25 个唯一用户必须进行至少 1000 次物品交互，每个用户至少有两次交互。为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

如果您不确定自己是否有足够的数据，可以使用 Amazon Personalize 控制台导入和分析数据。有关更多信息，请参阅 分析 Amazon Personalize 数据集中的数据质量和数量。

您是否有实时事件流式传输架构？

如果您没有足够的物品交互数据，则可以使用 Amazon Personalize 来收集其它实时事件数据。对于一些食谱和使用案例，Amazon Personalize 可以从用户的最新活动中学习，并在他们使用您的应用程序时更新建议。

有关记录事件的信息（包括事件如何影响建议、第三方事件跟踪服务列表以及示例实施），请参阅记录实时事件以影响推荐。

您的数据是否针对 Amazon Personalize 进行了优化？

我们建议您在数据中检查以下内容：

- 检查是否缺失值。我们建议至少 70% 的记录包含每个属性的数据。我们建议允许空值的列的填写率至少为 70%。
- 修复数据中的任何不准确之处或问题，例如命名约定不一致、项类别重复、数据集之间的 ID 不匹配或 ID 重复。这些问题可能会对建议产生负面影响或导致意外行为。例如，您的数据中可能同时有“N/A”和“不适用”，但仅基于“N/A”筛选建议。标记为“不适用”的物品不会被筛选器删除。

- 如果一个物品、用户或操作可以有多个类别（例如一部电影有多种类型），请将分类值合并到一个属性中，并使用 | 运算符分隔每个值。例如，电影的 GENRES 数据可能是动作 | 冒险 | 惊悚片。
- 避免为一列设置超过 1000 个可能的类别（除非该列包含仅用于筛选目的的数据）。

有关数据建议的完整列表以及如何使用 Amazon Personalize 识别问题的说明，请参阅分析 Amazon Personalize 数据集中的数据质量和数量。

您是否收集可以改进建议的可选数据？

以下数据可以帮助提高建议相关性。

- 事件类型（对所有域数据集组使用案例为必填项）
- 事件值
- 上下文元数据
- 物品和用户元数据
- 操作交互数据（仅由 PERSONALIZED_ACTIONS 配方使用）

有关 Amazon Personalize 可以使用的数据类型的更多信息，请参阅 Amazon Personalize 可以使用的数据类型。

您是否有计划测试您的建议？

您可以使用 A/B 测试来比较不同用户组与来自不同模型的建议进行交互的结果。A/B 测试可以帮助您比较不同的建议策略，并查看建议是否有助于您实现业务目标。有关更多信息，请参阅 通过 A/B 测试衡量推荐的影响。

您是否还有其他业务目标？

在一些情况下，除了为用户生成相关建议外，您可能还有其他目标。例如，您可能想要最大限度地提高收入，或者推广某个类别中某些类型的物品。以下 Amazon Personalize 特征可以提供帮助：

- 推广：您可以使用推广来确保一定比例的物品满足您的业务需求。有关更多信息，请参阅 通过实时推荐推广物品。
- 针对业务目标进行优化：对于某些自定义数据集组食谱，您可以针对自定义目标优化解决方案，例如最大限度地提高流式传输时长或增加收入。有关更多信息，请参阅 针对其他目标优化解决方案。

- 筛选建议。使用筛选器将业务规则应用于建议。您可以使用筛选器，在建议中包含或排除某些类型的物品。有关更多信息，请参阅 筛选建议和用户细分。

创建 Amazon Personalize 数据集组

在为数据创建架构 JSON 文件后，就可以创建数据集组。在 Amazon Personalize 中，数据集组 是 Amazon Personalize 资源（包括数据集、域推荐器和自定义资源）的容器。数据集组将资源组织到独立的集合中，在其中，一个数据集组中的资源不会影响任何其他数据集组中的资源。

您可以为每个业务域创建一个数据集组。例如，您可能有一个提供流视频的建议的应用程序，还有一个提供音频书籍的建议的应用程序。在 Amazon Personalize 中，您需要为每个应用程序创建一个数据集组。这样，来自一个应用程序的数据就不会影响 Amazon Personalize 为另一个应用程序生成的建议。

您可以创建域数据集组或自定义数据集组：

- 使用域数据集组，您可以创建针对不同使用案例进行预配置和优化的资源。创建数据集组时，通过指定 VIDEO_ON_DEMAND 或 ECOMMERCE 域，将其设置为域数据集组。

  如果您有流视频或电子商务应用程序，我们建议您创建域数据集组。您仍然可以添加自定义资源，例如针对自定义使用案例训练的解决方案和解决方案版本。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。
- 自定义数据集组 仅包含您根据使用案例配置的自定义资源。使用自定义资源，您可以根据业务需求，训练和部署可配置的解决方案与解决方案版本（经过训练的 Amazon Personalize 建议模型）。如果您没有 VIDEO_ON_DEMAND 或 ECOMMERCE 应用程序，我们建议您创建一个自定义数据集组。否则，我们建议从域数据集组开始，然后根据需要添加自定义资源。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 创建数据集组。

主题

- 创建数据集组（控制台）
- 创建数据集组 (AWS CLI)
- 创建数据集组 (AWS SDK)

创建数据集组（控制台）

通过在 Amazon Personalize 控制台中指定数据集组名称，创建数据集组。

创建数据集组

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 选择创建数据集组。
3. 如果这是您第一次使用 Amazon Personalize，请在创建数据集组页面的新建数据集组中，选择开始使用。
4. 在数据集组详细信息中，对于数据集组名称，为您的数据集组指定一个名称。
5. 选择您的域：
   - 选择电子商务，创建 ECOMMERCE 域数据集组。
   - 选择点播视频，创建 VIDEO_ON_DEMAND 域数据集组。
   - 选择自定义，创建仅包含自定义资源（例如解决方案、市场活动和批量推理作业）的自定义数据集组。
6. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
7. 选择创建数据集组。此时将显示概览页面。现在可以创建架构和数据集了。请参阅创建架构和数据集。

创建数据集组 (AWS CLI)

要创建数据集组，请使用 `create-dataset-group` 操作。要创建域数据集组，请为域指定 `ECOMMERCE` 或 `VIDEO_ON_DEMAND`。要创建自定义数据集组，请不要指定域。您可以使用标签参数，选择在 Amazon Personalize 中为资源添加标签。有关示例，请参阅添加标签 (AWS CLI)。

以下代码为 `VIDEO_ON_DEMAND` 域创建了域数据集组。

```
aws personalize create-dataset-group \
--name dataset-group-name \
--domain VIDEO_ON_DEMAND
```

如果成功，则数据集组 Amazon 资源名称 (ARN) 将显示，如下所示。

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

```
}
```

记录此值供将来使用。要显示您创建的数据集组，请使用 describe-dataset-group 命令，并指定返回的数据集组 ARN。

```
aws personalize describe-dataset-group \
--dataset-group-arn dataset group arn
```

数据集组及其属性显示如下。

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

当数据集组 status 为活动状态时，就可以创建架构和数据集了。请参阅创建架构和数据集。

创建数据集组 (AWS SDK)

使用以下代码创建域数据集组。为域数据集组命名，对于 domain，指定 ECOMMERCE 或 VIDEO_ON_DEMAND。要创建自定义数据集组，请修改代码以删除域参数。

有关 API 操作的更多信息，请参阅 API 参考部分的 CreateDatasetGroup。您可以使用标签参数，选择在 Amazon Personalize 中为资源添加标签。有关示例，请参阅添加标签 (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();
```

DescribeDatasetGroup 操作会返回操作的 datasetGroupArn 和状态。当数据集组 status 为活动状态时，就可以创建架构和数据集了。请参阅创建架构和数据集。

创建架构和数据集

创建数据集组后，就可以为要导入的每种数据类型创建 Amazon Personalize 架构和数据集了。架构会将您的数据结构告知给 Amazon Personalize，并允许 Amazon Personalize 解析数据。在 Amazon Personalize 中创建架构时，使用在为 Amazon Personalize 架构创建架构 JSON 文件中创建的 JSON 文件。

数据集是 Amazon Personalize 中用于训练数据的容器。不同的数据集类型有不同的要求。您可以为要导入的每种数据类型创建一个数据集。有关不同类型数据集以及如何准备数据的信息，请参阅为 Amazon Personalize 准备训练数据。

您可以使用 Amazon Personalize 控制台创建架构和数据集， AWS Command Line Interface (AWS CLI) 或 AWS SDKs。您无法在域数据集组中创建下一个最佳操作资源，包括操作和操作交互数据集。

> **Important**
>
> 创建架构后，您无法更改架构。但如果要添加新列，则可以将数据集的架构替换为新架构。有关更多信息，请参阅 替换数据集的架构，以便添加新列。

主题

- 创建数据集和架构（控制台）
- 创建数据集和架构 (AWS CLI)
- 创建数据集和架构 (AWS SDKs)

创建数据集和架构（控制台）

如果这是您的数据集组中的第一个数据集，则您的第一个数据集类型将是物品交互数据集。要在控制台中创建物品交互数据集，请指定数据集名称，然后以 Avro 格式指定 JSON 架构。如果这不是您在此数据集组中的第一个数据集，请选择数据集类型，然后指定名称和架构。

有关 Amazon Personalize 数据集要求的信息，请参阅为 Amazon Personalize 准备训练数据。如果您刚完成创建 Amazon Personalize 数据集组并且已经在创建数据集，请跳至此过程中的步骤 4。

创建数据集和架构

1. 如果还没有用于概述数据的架构 JSON 文件，请按照为 Amazon Personalize 架构创建架构 JSON 文件中的说明创建 JSON 文件。

2. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
3. 在数据集组页面上，选择您在创建 Amazon Personalize 数据集组中创建的数据集组。
4. 在步骤 1. 创建数据集和导入数据中，选择创建数据集，然后选择要创建的数据集类型。
5. 选择将数据直接导入 Amazon Personalize 数据集，然后选择下一步。
6. 在数据集详细信息中，对于数据集名称，为您的数据集指定名称。
7. 对于数据集架构，选择创建新架构或使用现有架构。
8. 如果您使用的是现有架构，请选择要使用的现有架构。如果要创建新架构，请为架构取名并粘贴到与您的数据匹配的架构 JSON 中。您在为 Amazon Personalize 架构创建架构 JSON 文件中创建了此文件。
9. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
10. 选择下一步，然后按照将训练数据导入 Amazon Personalize 数据集中的说明导入数据。

创建数据集和架构 (AWS CLI)

要使用创建数据集和架构 AWS CLI，请使用`create-schema`命令（使用 CreateSchema API 操作），然后使用命令`create-dataset`（使用 CreateDataset API 操作）。

创建架构和数据集

1. 如果还没有用于概述数据的架构 JSON 文件，请按照为 Amazon Personalize 架构创建架构 JSON 文件中的说明创建 JSON 文件。
2. 通过运行以下命令在 Amazon Personalize 中创建架构。创建架构后，您无法更改架构。将 `schemaName` 替换为架构的名称，将 `file://SchemaName.json` 替换为您的 JSON 文件的位置。此示例将该文件显示为属于当前文件夹。如果要为域数据集组中的数据集创建架构，则添加 `domain` 参数并将其设置为 `ECOMMERCE` 或 `VIDEO_ON_DEMAND`。有关 API 的更多信息，请参阅 CreateSchema。

   ```
   aws personalize create-schema \
     --name SchemaName \
     --schema file://SchemaName.json
   ```

   这将显示架构 Amazon 资源名称 (ARN)，如以下示例所示：

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

3. 通过运行以下命令来创建一个空数据集。提供来自创建数据集组 (AWS CLI) 的数据集组 Amazon 资源名称 (ARN) 和来自上一步的架构 ARN。数据集类型值可以是 `Interactions`、`Users`、`Items`、`Actions` 或 `Action_Interactions`。有关 API 的更多信息，请参阅 CreateDataset。

```
aws personalize create-dataset \
  --name Dataset Name \
  --dataset-group-arn Dataset Group ARN \
  --dataset-type Dataset Type \
  --schema-arn Schema Arn
```

这时将显示数据集 ARN，如以下示例所示。

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

4. 记录数据集 ARN 以备后用。创建数据集后，就可以导入训练数据了。请参阅将训练数据导入 Amazon Personalize 数据集。

创建数据集和架构 (AWS SDKs)

要使用创建数据集和架构 AWS SDKs，您需要先定义 Avro 格式的架构，然后使用CreateSchema操作将其添加到 A mazon Personalize 中。创建架构后，您无法更改架构。随后，使用 CreateDataset 操作创建数据集。

创建架构和数据集

1. 如果还没有用于概述数据的架构 JSON 文件，请按照为 Amazon Personalize 架构创建架构 JSON 文件中的说明创建 JSON 文件。
2. 使用以下代码在 Amazon Personalize 中创建架构。为架构指定名称，为架构 JSON 文件指定文件路径。如果要为域数据集组中的数据集创建架构，则添加 `domain` 参数并将其设置为 `ECOMMERCE` 或 `VIDEO_ON_DEMAND`。有关 API 的更多信息，请参阅 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 将返回新架构的 ARN。记录该 ARN，因为需要在下一步中使用。

3. 使用 CreateDataset 操作创建数据集。以下代码展示了如何创建数据集。指定数据集组的 Amazon 资源名称 (ARN)、上一步中的架构 ARN，并指定数据集类型。数据集类型值可以是 `Interactions`、`Users`、`Items`、`Actions` 或 `Action_Interactions`。有关不同类型数据集的信息，请参阅为 Amazon Personalize 准备训练数据。

   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();
```

创建数据集后，就可以导入训练数据了。请参阅[将训练数据导入 Amazon Personalize 数据集](#)。

将训练数据导入 Amazon Personalize 数据集

完成创建架构和数据集后，就可以将训练数据导入到数据集。导入数据时，您可以选择批量、单独导入记录（或两者兼而有之）。

- 批量导入涉及一次性导入大量历史记录。您可以自己准备批量数据，然后将其从 Amazon S3 中的 CSV 文件直接导入 Amazon Personalize。有关如何准备数据的信息，请参阅为 Amazon Personalize 准备训练数据。如果您在准备数据时需要帮助，可以使用 SageMaker AI Data Wrangler 来准备和导入批量项目交互、用户和项目数据。有关更多信息，请参阅 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。
- 如果您没有批量数据，则可以使用单独的导入操作来收集数据和直播事件，直到满足 Amazon Personalize 训练要求以及域应用场景或配方的数据要求。有关记录事件的信息，请参阅记录实时事件以影响推荐。有关导入单个记录的信息，请参阅将单个记录导入 Amazon Personalize 数据集。

将数据导入 Amazon Personalize 数据集后，您可以对其进行分析、将其导出到 Amazon S3 存储桶、对其进行更新，或者通过删除数据集将其删除。

如果您导入的项目、用户或操作的 ID 与您的数据集中已有的一条记录的 ID 相同，则 Amazon Personalize 会将该记录替换为新记录。如果您记录了两个具有完全相同时间戳和相同属性的项目交互或操作交互事件，则 Amazon Personalize 将只保留其中一个事件。

随着目录的增长，使用额外批量或单个数据导入操作来更新历史数据。要获得实时建议，请将您的物品交互数据集与用户的行为保持同步。为此，您可以使用事件跟踪器和 PutEvents 操作记录实时交互事件。有关更多信息，请参阅 记录实时事件以影响推荐。

导入数据后，就可以创建域推荐器（适用于域数据集组）或自定义资源（适用于自定义数据集组），以根据您的数据训练模型。使用这些资源生成建议。有关更多信息，请参阅 Amazon Personalize 中的域推荐器或 用于训练和部署 Amazon Personalize 模型的自定义资源。

主题

- 通过数据集导入作业将批量数据导入到 Amazon Personalize
- 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据
- 将单个记录导入 Amazon Personalize 数据集

通过数据集导入作业将批量数据导入到 Amazon Personalize

格式化输入数据（请参阅为 Amazon Personalize 准备训练数据）并完成创建架构和数据集后，就可以通过数据集导入作业导入批量数据。数据集导入作业是一种批量导入工具，可通过 Amazon S3 中的数据填充数据集。

要从 Amazon S3 导入数据，您的 CSV 文件必须在 Amazon S3 存储桶中，并且您必须向 Amazon Personalize 授予访问您的 Amazon S3 资源的权限：

- 有关将文件上传到 Amazon S3 的信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
- 若要了解如何向 Amazon Personalize 授予对 Amazon S3 中文件的访问权限，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

  如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

您可以使用 Amazon Personalize 控制台 AWS Command Line Interface （AWS CLI）或创建数据集导入任务 AWS SDKs。如果您之前为数据集创建了数据集导入作业，则可以使用新的数据集导入作业来添加或替换现有批量数据。有关更多信息，请参阅 训练后更新数据集中的数据。

如果您导入的项目、用户或操作的 ID 与您的数据集中已有的一条记录的 ID 相同，则 Amazon Personalize 会将该记录替换为新记录。如果您记录了两个具有完全相同时间戳和相同属性的项目交互或操作交互事件，则 Amazon Personalize 将只保留其中一个事件。

导入数据后，就可以创建域推荐器（适用于域数据集组）或自定义资源（适用于自定义数据集组），以根据您的数据训练模型。使用这些资源生成建议。有关更多信息，请参阅 Amazon Personalize 中的域推荐器或 用于训练和部署 Amazon Personalize 模型的自定义资源。

主题

- 导入模式
- 创建数据集导入作业（控制台）
- 创建数据集导入作业（AWS CLI）
- 创建数据集导入作业（AWS SDKs）

导入模式

如果您已经为数据集创建导入作业，则可以配置 Amazon Personalize 如何添加新记录。为此，您需要为数据集导入作业指定导入模式。如果您尚未导入批量记录，则导入模式字段在控制台中不可用，您只能在 `CreateDatasetImportJob` API 操作中指定 `FULL`。默认为完整替换。

- 要覆盖数据集中的所有现有批量数据，请在 Amazon Personalize 控制台中选择替换现有数据，或者在 CreateDatasetImportJob API 操作中指定 `FULL`。这不会取代您单独导入的数据，包括实时记录的事件。
- 要将记录附加到数据集中的现有数据，请选择添加到现有数据，或在 `CreateDatasetImportJob` API 操作中指定 `INCREMENTAL`。Amazon Personalize 会将所有具有相同 ID 的记录替换为新记录。

> **Note**
>
> 要使用数据集导入作业将数据附加到物品交互数据集或操作交互数据集，您必须至少有 1000 条新的物品交互或操作交互记录。

创建数据集导入作业（控制台）

> **Important**
>
> 默认情况下，数据集导入作业会替换您批量导入的数据集中的任何现有数据。如果您已经导入了批量数据，则可以通过更改作业的导入模式来追加数据。

要使用 Amazon Personalize 控制台将批量记录导入数据集，请创建一个包含名称、IAM 服务角色和数据位置的数据集导入作业。

如果您刚刚在创建架构和数据集中创建了数据集，请跳至步骤 5。

导入批量记录（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面，选择您的数据集组。这时，将显示数据集组概述。
3. 在导航窗格中，选择数据集，然后选择要将批量数据导入到其中的数据集。
4. 在数据集导入作业中，选择创建数据集导入作业。

5. 如果这是第一个数据集导入作业，则对于数据导入源，请选择从 S3 导入数据。
6. 对于数据集导入作业名称，为导入作业指定名称。
7. 如果您已导入批量数据，则对于导入模式，请选择如何更新数据集。选择替换现有数据或添加到现有数据。如果这是第一个数据集作业，则不会显示此选项。有关更多信息，请参阅 训练后更新数据集中的数据。
8. 在数据导入源中，对于数据位置，指定您的数据文件在 Amazon S3 中的存储位置。使用以下语法：

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

   如果您的 CSV 文件位于 Amazon S3 存储桶的某个文件夹中，并且您想通过一个数据集导入作业将多个 CSV 文件上传到一个数据集，则可以指定该文件夹的路径。Amazon Personalize 仅使用您文件夹第一级中的文件，不使用任何子文件夹中的任何数据。使用以下语法，在文件夹名称后面加上 / ：

   ```
   s3:/amzn-s3-demo-bucket/<folder path>/
   ```
9. 对于 IAM 角色，选择创建新角色或使用现有角色。如果您完成了先决条件，请选择使用现有服务角色，并指定您在为 Amazon Personalize 创建 IAM 角色中创建的角色。
10. 如果您创建了指标归因，并希望将与此作业相关的指标发布到 Amazon S3，请在将事件指标发布到 S3 中，选择发布此导入作业的指标。

    如果您尚未创建指标归因并想发布此作业的指标，请选择创建指标归因，以在其他选项卡上创建新的指标归因。创建指标归因后，您可以返回此屏幕，并完成导入作业的创建。

    有关指标归因的更多信息，请参阅衡量 Amazon Personalize 推荐的影响。
11. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
12. 选择 Start import (开始导入)。数据导入作业开始，并显示控制面板概览页。当状态显示为 ACTIVE 时，数据集导入完成。将数据导入 Amazon Personalize 数据集后，您可以分析数据、将数据导出到 Amazon S3 存储桶、更新数据，或通过删除数据集来删除数据。

    导入数据后，就可以创建域推荐器（适用于域数据集组）或自定义资源（适用于自定义数据集组），以根据您的数据训练模型。使用这些资源生成建议。有关更多信息，请参阅 Amazon Personalize 中的域推荐器或 用于训练和部署 Amazon Personalize 模型的自定义资源。

创建数据集导入作业（AWS CLI）

> **Important**
>
> 默认情况下，数据集导入作业会替换您批量导入的数据集中的任何现有数据。如果您已经导入了批量数据，则可以通过更改作业的导入模式来追加数据。

要使用导入批量记录 AWS CLI，请使用CreateDatasetImportJob命令创建数据集导入任务。如果您之前已经为数据集创建了数据集导入作业，则可以使用导入模式参数来指定如何添加新数据。有关更新现有批量数据的更多信息，请参阅训练后更新数据集中的数据。

导入批量记录 (AWS CLI)

1. 通过运行以下命令来创建数据集导入作业。为您的数据集提供 Amazon 资源名称 (ARN)，并指定存储训练数据的 Amazon S3 存储桶的路径。为该路径使用以下语法：

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

   如果您的 CSV 文件位于 Amazon S3 存储桶的某个文件夹中，并且您想通过一个数据集导入作业将多个 CSV 文件上传到一个数据集，则可以指定该文件夹的路径。Amazon Personalize 仅使用您文件夹第一级中的文件，不使用任何子文件夹中的任何数据。使用以下语法，在文件夹名称后面加上 /：

   s3:/amzn-s3-demo-bucket/<folder path>/

   提供您在中创建的 AWS Identity and Access Management (IAM) 角色亚马逊资源名称 (ARN)。为 Amazon Personalize 创建 IAM 角色默认 `import-mode` 为 `FULL`。有关更多信息，请参阅 训练后更新数据集中的数据。有关操作的更多信息，请参阅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
   ```

   数据集导入作业 ARN 将显示，如以下示例所示。

   ```
   {
   ```

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

2. 通过使用 describe-dataset-import-job 命令来检查状态。提供上一个步骤中返回的数据集导入作业 ARN。有关操作的更多信息，请参阅DescribeDatasetImportJob。

```
aws personalize describe-dataset-import-job \
--dataset-import-job-arn dataset import job arn
```

这将显示数据集导入作业的属性，包括其状态。最初，status 显示为 CREATE PENDING。

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

当状态显示为 ACTIVE 时，数据集导入完成。将数据导入 Amazon Personalize 数据集后，您可以分析数据、将数据导出到 Amazon S3 存储桶、更新数据，或通过删除数据集来删除数据。

导入数据后，就可以创建域推荐器（适用于域数据集组）或自定义资源（适用于自定义数据集组），以根据您的数据训练模型。使用这些资源生成建议。有关更多信息，请参阅 Amazon Personalize 中的域推荐器或 用于训练和部署 Amazon Personalize 模型的自定义资源。

创建数据集导入作业（AWS SDKs）

> **Important**
>
> 默认情况下，数据集导入作业会替换您批量导入的数据集中的任何现有数据。如果您已经导入了批量数据，则可以通过更改作业的导入模式来追加数据。

要导入数据，请通过 CreateDatasetImportJob 操作创建数据集导入作业。以下代码展示了如何创建数据集导入作业。

提供作业名称，将 `datasetArn` 设置为数据集的 Amazon 资源名称 (ARN)，并将 `dataLocation` 设置为存储训练数据的 Amazon S3 存储桶的路径。为该路径使用以下语法：

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

如果您的 CSV 文件位于 Amazon S3 存储桶的某个文件夹中，并且您想通过一个数据集导入作业将多个 CSV 文件上传到一个数据集，则可以指定该文件夹的路径。Amazon Personalize 仅使用您文件夹第一级中的文件，不使用任何子文件夹中的任何数据。使用以下语法，在文件夹名称后面加上 `/`：

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

对于`roleArn`，请指定授予 Amazon Personalize 访问您的 S3 存储桶权限的 AWS Identity and Access Management (IAM) 角色。请参阅为 Amazon Personalize 创建 IAM 角色。默认 `importMode` 为 `FULL`。这将替换数据集中的所有批量数据。要追加数据，请将其设置为 `INCREMENTAL`。有关更新现有批量数据的更多信息，请参阅训练后更新数据集中的数据。

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();
```

来自 DescribeDatasetImportJob 操作的响应包含操作的状态。

您必须等待，直到状态更改为 ACTIVE，然后才能使用数据训练模型。

当状态显示为 ACTIVE 时，数据集导入完成。将数据导入 Amazon Personalize 数据集后，您可以分析数据、将数据导出到 Amazon S3 存储桶、更新数据，或通过删除数据集来删除数据。

导入数据后，就可以创建域推荐器（适用于域数据集组）或自定义资源（适用于自定义数据集组），以根据您的数据训练模型。使用这些资源生成建议。有关更多信息，请参阅 Amazon Personalize 中的域推荐器或 用于训练和部署 Amazon Personalize 模型的自定义资源。

使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据

> **Important**
>
> 在使用 Data Wrangler 时，会产生 AI 成本。 SageMaker 有关费用和价格的完整列表，请参阅 A mazon A SageMaker I 定价的 Data Wrangler 选项卡。为避免产生额外费用，请在完成操作后关闭您的 Data Wrangler 实例。有关更多信息，请参阅关闭 Data Wrangler。

创建数据集组后，您可以使用 Amazon A SageMaker I Data Wrangler（Data Wrangler）将来自 40 多个来源的数据导入到 Amazon Personalize 数据集中。Data Wrangler 是 Amazon A SageMaker I Studio Classic 的一项功能，它为导入、准备、转换和分析数据提供了端到端解决方案。您不能使用 Data Wrangler 来准备数据并将其导入操作数据集或操作交互数据集。

使用 Data Wrangler 准备和导入数据时，您使用的是数据流。数据流 定义了一系列机器学习数据准备步骤，从导入数据开始。每次您向流添加步骤时，Data Wrangler 都会对您的数据执行操作，例如转换数据或生成可视化。

以下是您可以添加到流中用于为 Amazon Personalize 准备数据的一些步骤：

- 见解： 您可以将 Amazon Personalize 特定见解步骤添加到您的流中。这些见解有助于您了解数据及可采取哪些操作来改善数据。
- 可视化： 您可以添加可视化步骤来生成直方图和散点图等图表。图表可以帮助您发现数据中的问题，例如异常值或缺失值。
- 转换： 您可以使用特定于 Amazon Personalize 的一般转换步骤来确保您的数据符合 Amazon Personalize 要求。Amazon Personalize 转换可帮助您将数据列映射到所需的列，具体取决于 Amazon Personalize 数据集类型。

如果您需要在将数据导入 Amazon Personalize 之前退出 Data Wrangler，则可以在从 Amazon Personalize 控制台启动 Data Wrangler 时选择相同的数据集类型，返回您之前离开的位置。或者你可以直接通过 SageMaker AI Studio Classic 访问 Data Wrangler。

我们建议您将数据从 Data Wrangler 导入到 Amazon Personalize，如下所示。转换、可视化和分析步骤是可选、可重复的步骤，且可以按任何顺序完成。

1. 设置权限-为 Amazon Personalize 和 SageMaker AI 服务角色设置权限。而且，为您的用户设置权限。

2. 从 Amazon Personalize 控制台在 SageMaker AI Studio Classic 中启动 Data Wrangler — 使用亚马逊个性化控制台配置 A SageMaker I 域并在 AI Studio Classic 中 SageMaker 启动 Data Wrangler。
3. 将您的数据导入 Data Wrangler - 将来自 40 多个源的数据导入 Data Wrangler。来源包括诸如亚马逊 Redshift、亚马逊 EMR 或亚马逊 Athena 之类的 AWS 服务，以及第三方，例如 Snowflake 或。DataBricks
4. 转换您的数据 - 使用 Data Wrangler 转换您的数据，以满足 Amazon Personalize 要求。
5. 可视化和分析您的数据 - 使用 Data Wrangler 可视化您的数据，并通过 Amazon Personalize 特定见解对其进行分析。
6. 处理数据并将其导入 Amazon Personalize — 使用 A SageMaker I Studio Classic Jupyter 笔记本将处理过的数据导入亚马逊个性化。

附加信息

以下资源提供了有关使用亚马逊 A SageMaker I Data Wrangler 和 Amazon Personalize 的更多信息。

- 有关指导您处理和转换示例数据集的教程，请参阅 A ma SageMaker zon AI 开发者指南中的演示：Data Wrangler Titanic 数据集演练。本教程介绍了 Data Wrangler 的字段和函数。
- 有关注册亚马逊 A SageMaker I 域的信息，请参阅《亚马逊 AI 开发者指南》中的 “快速加入亚马逊 A SageMaker SageMaker I 域名”。
- 有关 Amazon Personalize 数据要求的信息，请参阅为 Amazon Personalize 准备训练数据。

设置权限

要使用 Data Wrangler 准备数据，必须设置以下权限：

- 为 Amazon Personalize 创建服务角色： 如果您尚未创建，请按照设置 Amazon Personalize 中的说明，为 Amazon Personalize 创建 IAM 服务角色。此角色必须对存储处理过数据的 Amazon S3 存储桶具有 `GetObject` 和 `ListBucket` 权限。而且它必须拥有使用任何 AWS KMS 密钥的权限。

  有关向 Amazon Personalize 授予对 Amazon S3 存储桶的访问权限的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。有关授予 Amazon Personalize 访问您的 AWS KMS 密钥的权限的信息，请参阅授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
- 创建具有 SageMaker AI 权限的管理用户：您的管理员必须拥有对 SageMaker AI 的完全访问权限，并且必须能够创建 A SageMaker I 域。有关更多信息，请参阅 Amazon A SageMaker I 开发者指南中的创建管理用户和群组。

- 创建 A SageMaker I 执行角色：创建可访问 A SageMaker SageMaker I 资源和 Amazon Personalize 数据导入操作的人工智能执行角色。A SageMaker I 执行角色必须附加AmazonSageMakerFullAccess策略。如果您需要更精细的 Data Wrangler 权限，请参阅 Amazon AI 开发者指南中的 Data Wrangler 安全和权限。 SageMaker 有关 SageMaker AI 角色的更多信息，请参阅 SageMaker AI 角色。

  要授予对 Amazon Personalize 数据导入操作的访问权限，请将以下 IAM 策略附加到 A SageMaker I 执行角色。该策略会授予将数据导入 Amazon Personalize 并将策略附加到 Amazon S3 存储桶所需的权限。而且，当服务为 Amazon Personalize 时，它会授予 PassRole 权限。使用 Data Wrangler 准备格式化数据后，将 Amazon S3 amzn-s3-demo-bucket 更新为您要用作此类数据目标的 Amazon S3 存储桶的名称。

  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"
                }
            }
        }
    ]
}
```

有关创建 IAM 策略的信息，请参阅《IAM 用户指南》中的创建 IAM 策略。有关将 IAM 策略附加到角色的信息，请参阅《IAM 用户指南》中的添加和删除 IAM 身份权限。

从 Amazon Personalize 启动 Data Wrangler

要从 Amazon Personalize 启动 Data Wrangler，你可以使用 Amazon Personalize 控制台配置 SageMaker AI 域并启动 Data Wrangler。

从 Amazon Personalize 启动 Data Wrangler

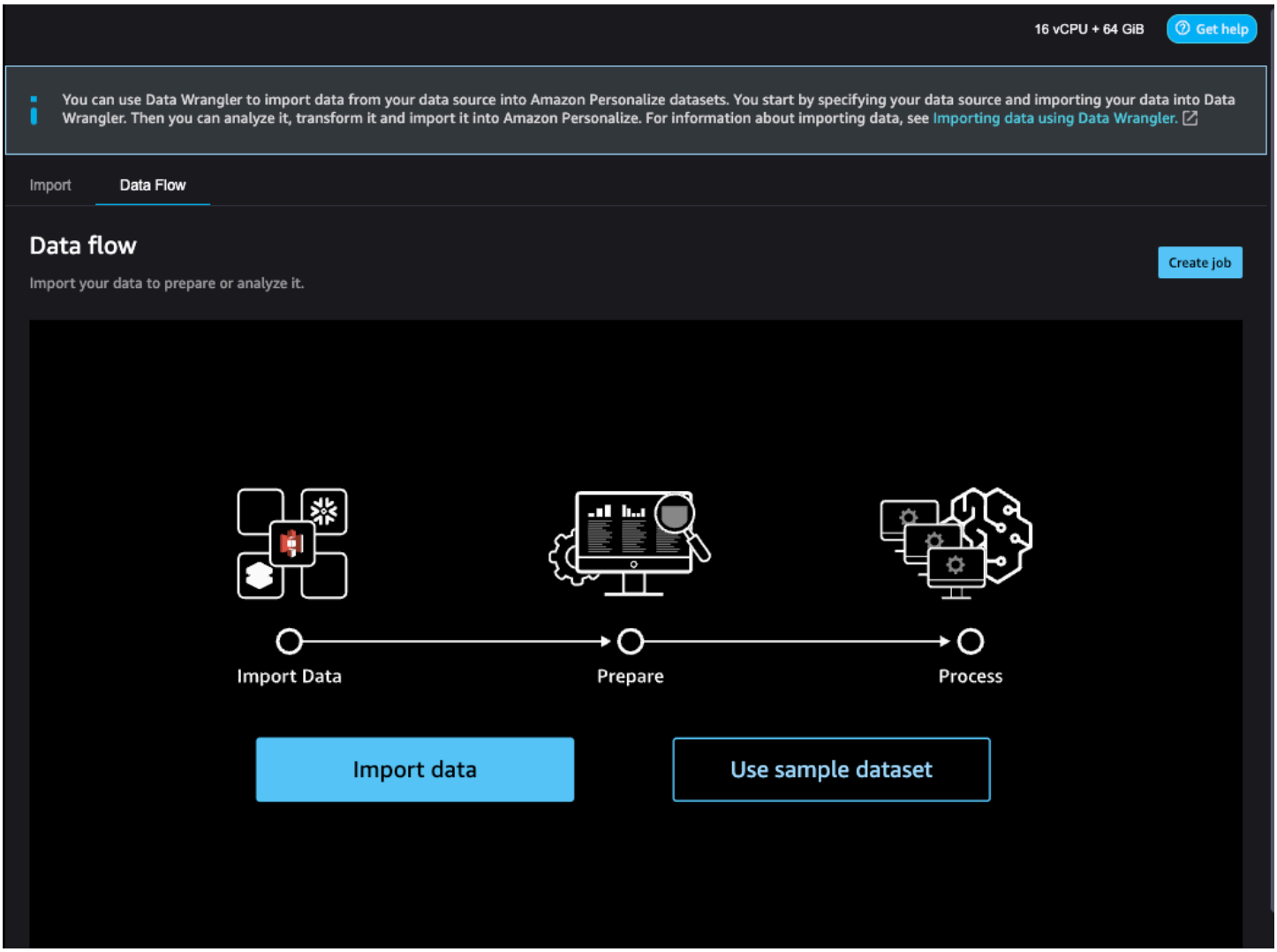
1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 在设置数据集中，选择创建数据集并选择要创建的数据集类型。您不能使用 Data Wrangler 来准备操作数据集或操作交互数据集。
4. 选择使用 Data Wrangler 导入数据，然后选择下一步。
5. 对于SageMaker 域，请选择使用现有域或创建新域。你需要一个 SageMaker AI 域才能在 A SageMaker I Studio Classic 中访问 Data Wrangler。有关域和用户资料的信息，请参阅 SageMaker A mazon AI 开发者指南中的 SageMaker AI 域。
6. 要使用现有域，请选择 A SageMaker I 域和用户配置文件来配置该域。
7. 创建新域：
   - 为新域命名。
   - 选择用户配置文件名称。
   - 对于执行角色，选择您在设置权限中创建的角色。或者，如果您有 CreateRole 权限，请使用角色创建向导创建新角色。您使用的角色必须附加 `AmazonSageMakerFullAccess` 策略。
8. 选择下一步。如果您要创建新域， SageMaker AI 会开始创建您的域名。这一过程耗时最多 10 分钟。

9. 查看您的 SageMaker AI 域的详细信息。
10. 选择 “使用 Data Wrangler 导入数据”。 SageMaker AI Studio Classic 开始创建您的环境，完成后，A SageMaker I Studio Classic 中的 Data Wrangler 的数据流页面将在新选项卡中打开。SageMaker AI Studio Classic 最多可能需要五分钟才能完成环境的创建。完成后，您就可以开始将数据导入 Data Wrangler 了。有关更多信息，请参阅 将数据导入 Data Wrangler。

将数据导入 Data Wrangler

配置 SageMaker AI 域并在新选项卡中启动 Data Wrangler 后，就可以将数据从源导入到 Data Wrangler 了。当您使用 Data Wrangler 为 Amazon Personalize 准备数据时，一次只能导入一个数据集。我们建议从物品交互数据集开始。您不能使用 Data Wrangler 来准备操作数据集或操作交互数据集。

您从数据流页面开始。该页面应类似于以下内容。



要开始导入数据，请选择导入数据并指定您的数据源。Data Wrangler 支持 40 多个源。其中包括诸如亚马逊 Redshift、亚马逊 EMR 或亚马逊 Athena 之类的 AWS 服务，以及第三方，例如 Snowflake 或。 DataBricks不同的数据源有不同的连接和导入数据的步骤。

有关可用来源的完整列表和有关导入数据的分步说明，请参阅 Amazon A SageMaker I 开发者指南中的导入。

将数据导入 Data Wrangler 后，就可以对其进行转换了。有关数据转换的信息，请参阅转换数据。

转换数据

要在 Data Wrangler 中转换数据，请在数据流中添加转换步骤。Data Wrangler 包含 300 多种转换，可用于准备数据，包括为 Amazon Personalize 映射列转换。而且，您可以使用常规 Data Wrangler 转换来修复异常值、类型问题和缺失值等问题。

完成数据转换后，您可以使用 Data Wrangler 对其进行分析。或者，如果您在 Data Wrangler 中完成了数据的准备，则可以对其进行处理并将其导入到 Amazon Personalize 中。有关分析数据的信息，请参阅生成可视化和数据见解。有关处理和导入数据的信息，请参阅处理数据并将其导入 Amazon Personalize。

主题

- 为 Amazon Personalize 映射列
- 常规 Data Wrangler 转换

为 Amazon Personalize 映射列

要转换数据使其符合 Amazon Personalize 要求，您需添加为 Amazon Personalize 映射列转换，并将列映射到 Amazon Personalize 的必填和可选字段。

使用为 Amazon Personalize 映射列转换

1. 选择最新转换对应的 +，然后选择添加转换。如果您尚未添加转换，请为数据类型转换选择 +。Data Wrangler 会自动将此转换添加到您的流中。
2. 选择添加步骤。
3. 选择 Amazon Personalize 的转换。默认情况下，为 Amazon Personalize 映射列转换处于选中状态。
4. 使用转换字段，将您的数据映射到所需的 Amazon Personalize 属性。

    1. 选择与您的数据相匹配的数据集类型（交互、物品或用户）。

2. 选择您的域（ECOMMERCE、VIDEO_ON_DEMAND 或自定义）。您选择的域必须与您在创建数据集组时指定的域相匹配。
3. 选择与 Amazon Personalize 必填字段和可选字段相匹配的列。例如，对于 item_ID 列，选择数据中存储每件物品唯一标识信息的列。

   每个列字段都按数据类型进行筛选。您的数据中只有符合 Amazon Personalize 数据类型要求的列才可用。如果您的数据不是所需的类型，则您可以使用将值解析为类型 Data Wrangler 转换，对其进行转换。

常规 Data Wrangler 转换

以下常规 Data Wrangler 转换可以帮助您为 Amazon Personalize 准备数据：

- 数据类型转换：如果您的字段未作为可能的选项列在为 Amazon Personalize 映射列转换中，则您可能需要转换其数据类型。Data Wrangler 转换将值解析为类型可以帮助您转换数据。或者您可以使用 Data Wrangler 在创建流时默认添加的数据类型转换。要使用此转换，请从类型下拉列表中选择数据类型，选择预览，然后选择更新。

  有关字段必需的数据类型的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件中与您的域和数据集类型对应的部分。
- 处理缺失值和异常值：如果您生成缺失值或异常值见解，则可以使用 Data Wrangler 转换处理异常值和处理缺失值来解决这些问题。
- 自定义转换：使用 Data Wrangler，你可以使用 PythonUser-Defined （函数） PySpark、pandas 或 (SQL) 创建自己的转换。 PySpark 您可以使用自定义转换来执行诸如删除重复列或按列分组之类的任务。有关更多信息，请参阅 Amazon A SageMaker I 开发者指南中的自定义转换。

生成可视化和数据见解

将数据导入 Data Wrangler 后，您可以使用它来生成可视化和数据见解。

- 可视化： Data Wrangler 可以生成不同类型的图表，例如直方图和散点图。例如，您可以生成直方图来识别数据中的异常值。
- 数据见解： 您可以使用适用于 Amazon Personalize 的数据质量和见解报告，通过数据见解以及列和行统计信息来了解您的数据。此报告可以告知您数据中是否存在任何类型问题。您还可以了解可采取哪些操作来改善数据。这些操作可以帮助您满足 Amazon Personalize 资源要求，例如模型训练要求，也可以改进建议。

通过可视化和见解了解数据后，您可以使用这些信息帮助您应用其他转换来改善数据。或者，如果您完成了数据准备，则可以对其进行处理并将其导入到 Amazon Personalize 中。有关数据转换的信息，请参阅转换数据。有关处理和导入数据的信息，请参阅处理数据并将其导入 Amazon Personalize。

生成可视化

您可以使用 Data Wrangler 创建不同类型的图表，例如直方图和散点图。例如，您可以生成直方图来识别数据中的异常值。要生成数据可视化，请在流中添加分析步骤，然后从分析类型中选择要创建的可视化。

有关在 Data Wrangler 中创建可视化的更多信息，请参阅 A mazon A SageMaker I 开发者指南中的分析和可视化。

生成数据见解

您可以使用 Data Wrangler，生成特定于数据集类型的 Amazon Personalize 的数据质量和见解报告。在生成报告之前，我们建议您转换数据以满足 Amazon Personalize 的要求。这将生成更多相关的见解。有关更多信息，请参阅 转换数据。

主题

- 报告内容
- 生成报告

报告内容

Amazon Personalize 的数据质量和见解报告 包括以下部分：

- 摘要： 报告摘要包括数据集统计信息和高优先级警告：
  - 数据集统计信息： 包括 Amazon Personalize 的特定统计信息（例如交互数据中的唯一用户数量）以及一般统计信息（例如缺失值或异常值的数量）。
  - 高优先级警告： 这些是 Amazon Personalize 的特定见解，对训练或建议的影响最大。每个警告都包含可用于解决问题的建议操作。
- 重复行和不完整行： 这些部分包含有关数据中哪些行存在缺失值以及哪些行重复的信息。
- 特征摘要： 本部分包括每列的数据类型、无效或缺失的数据信息以及警告计数。
- 特征详情： 本部分包括含每列数据详细信息的子部分。每个子部分都包含列的统计信息，例如类别值计数和缺失值信息。而且，每个子部分都包含 Amazon Personalize 针对数据列的具体见解和建议的操作。例如，见解可能表明某一列包含超过 30 个可能的类别。

数据类型问题

报告会识别数据类型不正确的列，并指定所需的类型。要获得与这些特征相关的见解，您必须转换列的数据类型并重新生成报告。要转换类型，您可以使用 Data Wrangler 转换将值解析为类型。

Amazon Personalize 见解

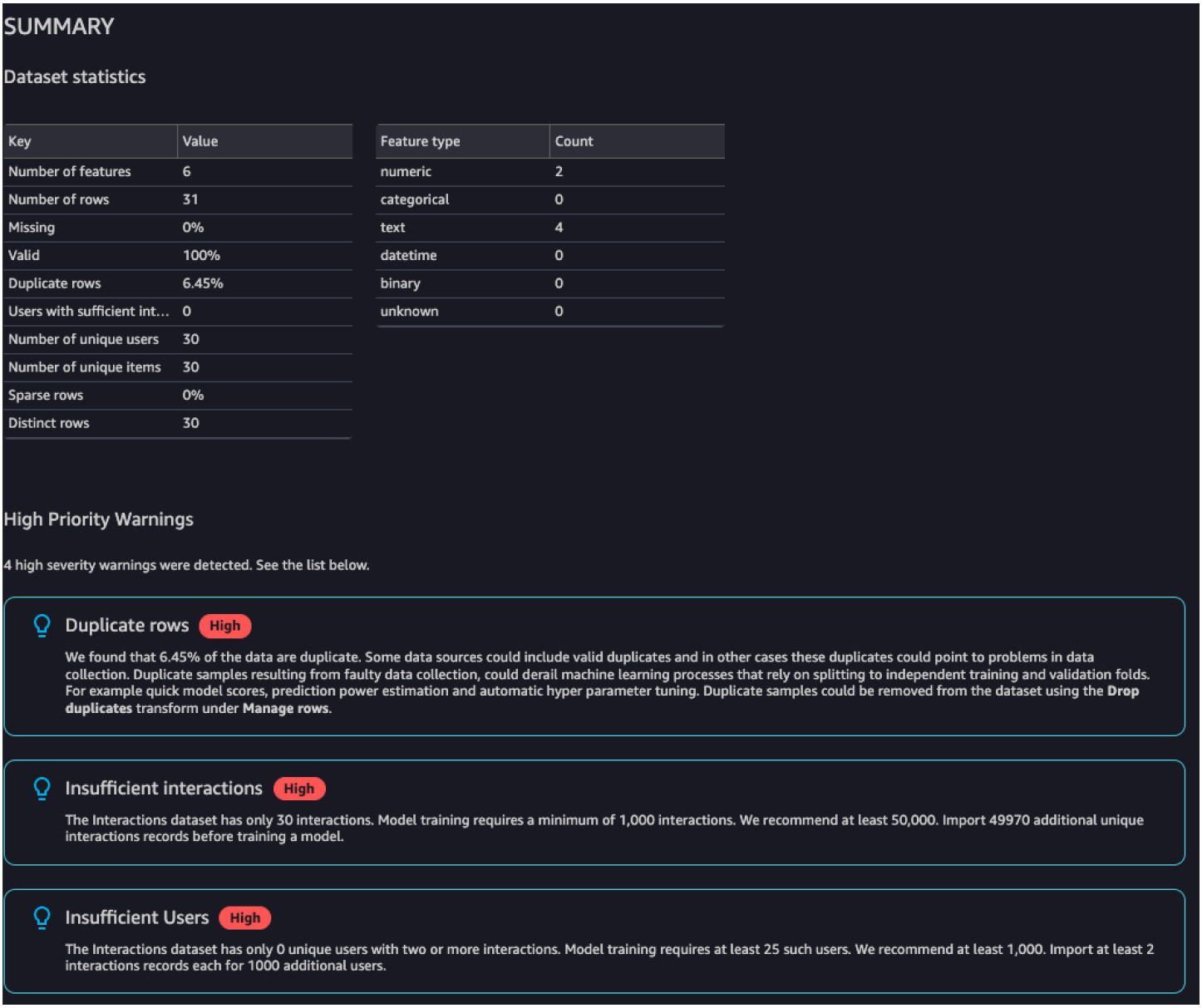
Amazon Personalize 见解包括调查结果和建议的操作。操作是可选的。例如，报告可能包含与一列分类数据的类别数量相关的见解和操作。如果您认为该列不是类别列，则可以忽略此见解，不采取任何操作。

除了细微的措辞差异外，Amazon Personalize 的特定见解与您在使用 Amazon Personalize 分析数据时可能生成的单个数据集 见解相同。例如，Data Wrangler 中的见解报告包含诸如“物品交互数据集只有 X 个具有两次或更多次交互的唯一用户”之类的见解。但它不包括诸如“物品数据集 中有 X% 的物品在物品交互数据集中没有交互”之类的见解。

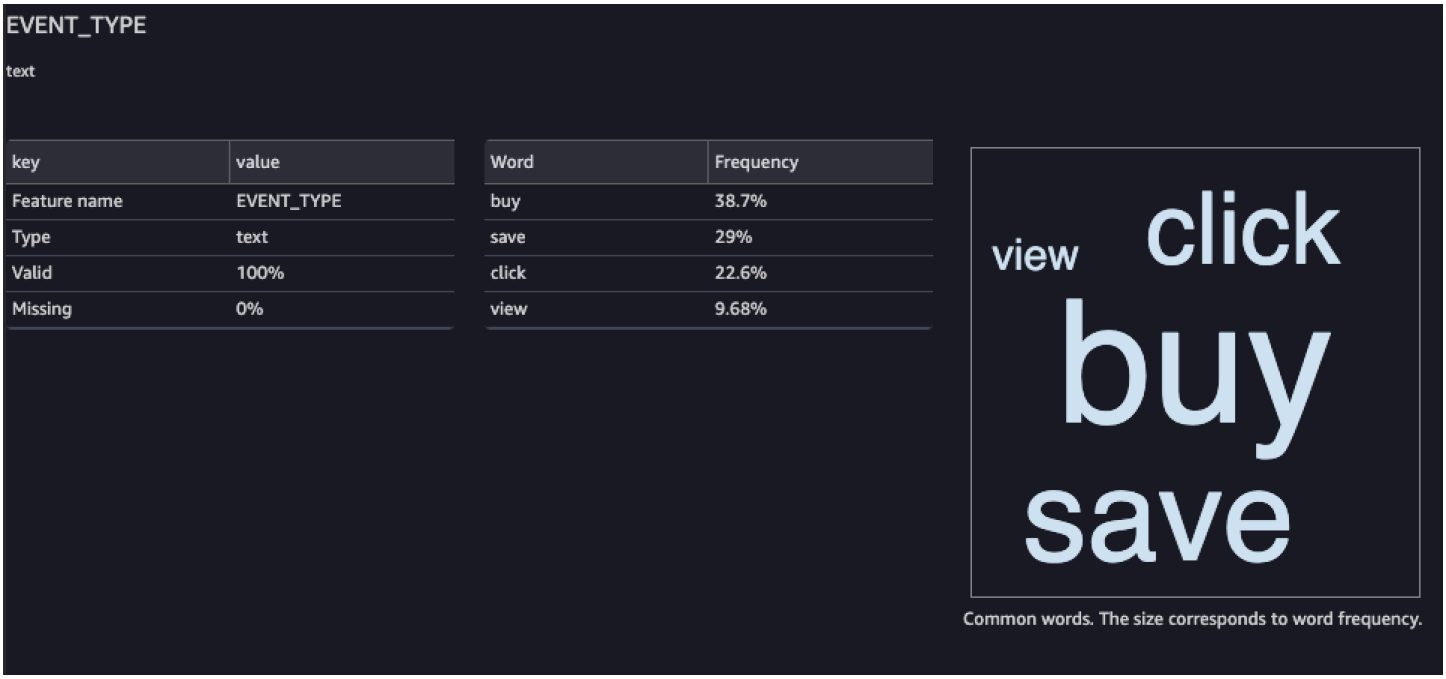
有关可能的 Amazon Personalize 特定见解列表，请参阅数据见解中未引用多个数据集的见解。

报告示例

Amazon Personalize 报告的外观和风格与 Data Wrangler 中的一般见解报告相同。有关一般见解报告的示例，请参阅《Amazon A SageMaker I 开发者指南》中的 “获取有关数据和数据质量的见解”。以下示例显示了物品交互数据集报告的摘要部分。它包括数据集统计信息和一些可能的高优先级物品交互数据集警告。



以下示例显示了物品交互数据集的 EVENT_TYPE 列的特征详细信息部分可能在报告中的显示方式。



生成报告

要生成 Amazon Personalize 的数据质量和见解报告，请为您的转换选择获取数据见解，然后创建分析。

生成 Amazon Personalize 的数据质量和见解报告

1. 为您要分析的转换选择 + 选项。如果您尚未添加转换，请为数据类型转换选择 +。Data Wrangler 会自动将此转换添加到您的流中。
2. 选择获取数据见解。这时，将显示创建分析面板。
3. 对于分析类型，选择 Amazon Personalize 的数据质量和见解报告。
4. 对于数据集类型，选择您要分析的 Amazon Personalize 数据集的类型。
5. （可选）选择针对完整数据运行。默认情况下，Data Wrangler 仅生成有关数据样本的见解。
6. 选择创建。分析完成后，将显示报告。

处理数据并将其导入 Amazon Personalize

当您完成数据分析和转换后，就可以对其进行处理并将其导入 Amazon Personalize 了。

- 处理数据 - 处理数据会将转换应用于整个数据集并将其输出到您指定的目的地。在这种情况下，您可以指定 Amazon S3 存储桶。

- 将数据导入 Amazon P ersonalize — 要将处理过的数据导入亚马逊个性化，您需要运行 A SageMaker I Studio Classic 中提供的 Jupyter 笔记本。此笔记本可创建您的 Amazon Personalize 数据集并将您的数据导入其中。

处理数据

在将数据导入 Amazon Personalize 之前，您必须将转换应用于整个数据集并将其输出到 Amazon S3 存储桶。为此，您需要创建一个目标节点，并将目标设置为 Amazon S3 存储桶，然后启动转换的处理作业。

有关指定目标和启动流程任务的分步说明，请参阅使用 Amazon A SageMaker I Data Wrangler 点击几下即可启动处理任务。添加目标时，选择 Amazon S3。将处理过的数据导入 Amazon Personalize 时，您将使用此位置。

处理完数据后，您就可以将其从 Amazon S3 存储桶导入到 Amazon Personalize 了。

将数据导入 Amazon Personalize

处理完数据后，即可将其导入 Amazon Personalize。要将处理后的数据导入 Amazon Personalize，您需要运行 A SageMaker I Studio Classic 中提供的 Jupyter 笔记本。此笔记本可创建您的 Amazon Personalize 数据集并将您的数据导入其中。

将处理过的数据导入 Amazon Personalize

1. 对于要导出的转换，选择导出到，然后选择 Amazon Personalize（通过 Jupyter 笔记本）。
2. 修改笔记本，以指定您用作处理作业数据目标的 Amazon S3 存储桶。（可选）为您的数据集组指定域。默认情况下，笔记本会创建自定义数据集组。
3. 查看创建架构的笔记本单元格。在运行单元格之前，验证架构字段是否具有预期的类型和属性。

   - 验证支持空数据的字段是否在类型列表中列出了 `null`。以下示例说明了如何为字段添加 `null`。

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
}
```

- 验证分类字段的类别属性是否设置为 true。以下示例说明了如何标记字段类别。

```
{
          "name": "SUBSCRIPTION_MODEL",
          "type": "string",
          "categorical": true
}
```

- 验证文本字段的文本属性是否设置为 true。以下示例说明了如何将字段标记为文本。

```
{
      "name": "DESCRIPTION",
      "type": [
        "null",
        "string"
      ],
      "textual": true
}
```

4. 运行笔记本以创建架构，创建数据集，然后将您的数据导入 Amazon Personalize 数据集。你可以像 SageMaker AI Studio Classic 之外的笔记本电脑一样运行笔记本电脑。有关运行 Jupyter 笔记本的信息，请参阅运行代码。有关 SageMaker AI Studio Classic 中笔记本电脑的信息，请参阅亚马逊 SageMaker AI 开发者指南中的使用亚马逊 SageMaker AI 笔记本电脑。

   完成笔记本后，如果您导入了交互数据，则可以创建推荐器或自定义资源。或者，您可以对物品数据集或用户数据集重复该过程。

   - 有关创建域推荐器的信息，请参阅Amazon Personalize 中的域推荐器。
   - 有关创建和部署自定义资源的信息，请参阅用于训练和部署 Amazon Personalize 模型的自定义资源。

将单个记录导入 Amazon Personalize 数据集

完成创建架构和数据集之后，可以将单个记录（包括物品交互、用户、物品、操作或操作交互）导入到现有数据集中。单独导入数据允许您随着目录的增长将小批量记录添加到您的 Amazon Personalize 数据集中。每次导入操作最多可以导入 10 条记录。

如果您导入的项目、用户或操作的 ID 与您的数据集中已有的一条记录的 ID 相同，则 Amazon Personalize 会将该记录替换为新记录。如果您记录了两个具有完全相同时间戳和相同属性的项目交互或操作交互事件，则 Amazon Personalize 将只保留其中一个事件。

如果您使用 Apache Kafka，则可以通过 Amazon Personalize Kafka 连接器 将数据实时流式传输到 Amazon Personalize。有关信息，请参阅 personalize-kafka-connector Github 存储库中的 Amazon Personalize Kafka 连接器。

如果您有大量历史记录，我们建议您先批量导入数据，然后根据需要单独导入数据。请参阅通过数据集导入作业将批量数据导入到 Amazon Personalize。

单个记录导入的筛选器更新

Amazon Personalize 会在上次单独导入后的 20 分钟内使用您的新交互、物品和用户数据更新您在数据集组中创建的所有筛选器。此更新允许您的市场活动在筛选针对用户的建议时使用最新数据。

如果您已经创建了推荐器或通过市场活动部署了自定义解决方案版本，则新单独记录如何影响建议取决于您使用的域使用案例或食谱。有关更多信息，请参阅 训练后更新数据集中的数据。

主题

- 单独导入交互
- 单独导入用户
- 单独导入物品
- 单独导入操作

单独导入交互

完成创建架构和数据集以创建物品交互数据集后，您可以单独将一个或多个新事件导入数据集。要单独导入交互事件，您需创建一个事件跟踪器，然后将一个或多个事件导入您的物品交互数据集。您可以使用 Amazon Personalize 控制台导入历史个人互动事件，也可以使用 AWS Command Line Interface (AWS CLI) 或导入历史或实时事件 AWS SDKs。

本部分包含有关使用 Amazon Personalize 控制台导入事件的信息。我们建议使用 Amazon Personalize 控制台仅 导入历史事件。有关使用 AWS CLI 或实时记录事件的信息，请参阅记录实时事件以影响推荐。 AWS SDKs

有关 Amazon Personalize 如何更新新记录筛选器以及新记录如何影响建议的信息，请参阅将单个记录导入 Amazon Personalize 数据集。

主题

- 创建事件跟踪器（控制台）
- 单独导入事件（控制台）

创建事件跟踪器（控制台）

> **Note**
>
> 如果您创建了事件跟踪器，则可以跳至单独导入事件（控制台）。

在将事件导入交互数据集之前，您必须为该数据集组创建事件跟踪器。

创建事件跟踪器（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要将事件导入到的物品交互数据集的数据集组。
3. 在数据集组的控制面板上，在安装事件摄取 SDK 中，选择开始。
4. 在配置跟踪器页面的跟踪器配置中，对于跟踪器名称，提供事件跟踪器的名称，然后选择下一步。
5. “安装 SDK” 页面显示了新事件跟踪器的跟踪 ID 以及使用 AWS Amplify 或 AWS Lambda 流式传输事件数据的说明。

   您可以忽略这些信息，因为您使用的是 Amazon Personalize 控制台来上传事件数据。如果您想 AWS Lambda 在将来使用 AWS Amplify 或流式传输事件数据，则可以在事件跟踪器页面上选择事件跟踪器来查看此信息。
6. 选择结束。现在，您可以使用控制台导入事件（参见单独导入事件（控制台））或使用 `PutEvents` 操作实时记录事件（参阅记录实时事件以影响推荐）。

单独导入事件（控制台）

创建事件跟踪器后，您可以将事件单独导入物品交互数据集。此过程假定您已经创建了一个物品交互数据集。有关创建数据集的信息，请参阅创建架构和数据集。

单独导入事件（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要将事件导入到的物品交互数据集的数据集组。
3. 在导航窗格中，选择数据集。
4. 在数据集页面上，选择交互数据集。

5. 在数据集详细信息页面的右上角，选择修改数据集，然后选择创建记录。
6. 在创建用户-物品交互记录页面中，对于记录输入，以 JSON 格式输入事件详细信息。事件的字段名称和值必须与您在创建物品交互数据集时使用的架构匹配。Amazon Personalize 提供一个 JSON 模板，其中包含该架构中的字段名称和数据类型。您一次最多可以导入 10 个事件。
7. 选择创建记录。在响应中，将列出导入结果并显示成功或失败消息。

单独导入用户

完成创建架构和数据集以创建用户数据集后，您可以单独将一个或多个新用户导入数据集。单独导入用户允许您随着目录的增长而进行小批量导入，从而使您的用户数据集保持最新状态。您一次最多可以导入 10 个用户。如果您有大量新用户，我们建议您先批量导入数据，然后根据需要单独导入用户数据。请参阅通过数据集导入作业将批量数据导入到 Amazon Personalize。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs 导入用户。如果您导入的用户的 `userId` 与您用户数据集中已有的用户 ID 相同，则 Amazon Personalize 会将该用户替换为新用户。您一次最多可以导入 10 个用户。

有关 Amazon Personalize 如何更新新记录筛选器以及新记录如何影响建议的信息，请参阅将单个记录导入 Amazon Personalize 数据集。

主题

- 单独导入用户（控制台）
- 单独导入用户 (AWS CLI)
- 单独导入用户 (AWS SDKs)

单独导入用户（控制台）

您一次最多可以导入 10 个用户。此过程假定您已创建一个用户数据集。有关创建数据集的信息，请参阅创建架构和数据集。

单独导入用户（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要将用户导入到的用户数据集的数据集组。
3. 在导航窗格中，选择数据集。

4. 在数据集页面上，选择用户数据集。
5. 在数据集详细信息页面的右上角，选择修改数据集，然后选择创建记录。
6. 在创建用户记录页面中，对于记录输入，以 JSON 格式输入用户详细信息。用户的字段名称和值必须与您在创建用户数据集时使用的架构匹配。Amazon Personalize 提供一个 JSON 模板，其中包含该架构中的字段名称和数据类型。
7. 选择创建记录。在响应中，将列出导入结果并显示成功或失败消息。

单独导入用户 (AWS CLI)

使用 PutUsers 操作将一个或多个用户添加到您的用户数据集。通过单次 PutUsers 调用最多可以导入 10 个用户。此部分假定您已经创建了一个用户数据集。有关创建数据集的信息，请参阅创建架构和数据集。

通过 AWS CLI使用以下 put-users 命令添加一个或多个用户。将 dataset arn 替换为数据集的 Amazon 资源名称 (ARN)，将 user Id 替换为用户 ID。如果具有相同 userId 的用户已存在于用户数据集中，则 Amazon Personalize 会将其替换为新用户。

对于 properties，针对用户数据集中的每个字段，将 propertyName 替换为架构中的字段名称（采用混合大小写）。例如，GENDER 将为 gender，MEMBERSHIP_TYPE 将为 membershipType。将 user data 替换为用户的数据。对于分类字符串数据，要包含单个属性的多个类别，请使用竖线 (|) 分隔每个类别。例如 \"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\"}"
    }]'
```

单独导入用户 (AWS SDKs)

使用 PutUsers 操作将一个或多个用户添加到您的用户数据集。如果具有相同 userId 的用户已存在于用户数据集中，则 Amazon Personalize 会将其替换为新用户。通过单次 PutUsers 调用最多可以导入 10 个用户。此部分假定您已创建一个用户数据集。有关创建数据集的信息，请参阅创建架构和数据集。

以下代码显示如何将一个或多个用户添加到您的用户数据集。对于每个属性名称参数，传递架构中的字段名称（采用混合大小写）。例如，GENDER 将为 gender，MEMBERSHIP_TYPE 将为 membershipType。对于每个属性值参数，传递用户数据。

对于分类字符串数据，要包含单个属性的多个类别，请使用竖线 (|) 分隔每个类别。例如 "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();
```

单独导入物品

完成创建架构和数据集以创建物品数据集后，您可以单独将一个或多个新物品导入数据集。单独导入物品允许您随着目录的增长而进行小批量导入，从而使您的物品数据集保持最新状态。您一次最多可以导入 10 个物品。如果您有大量新物品，我们建议您先批量导入数据，然后根据需要单独导入物品数据。请参阅通过数据集导入作业将批量数据导入到 Amazon Personalize。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs 导入商品。如果您导入的物品的 `itemId` 与您物品数据集中已有的物品 ID 相同，则 Amazon Personalize 会将其替换为新物品。

有关 Amazon Personalize 如何更新新记录筛选器以及新记录如何影响建议的信息，请参阅将单个记录导入 Amazon Personalize 数据集。

主题

- 单独导入物品（控制台）

- 单独导入物品 (AWS CLI)
- 单独导入物品 (AWS SDKs)

单独导入物品（控制台）

您一次最多可以向物品数据集导入 10 个物品。此过程假定您已经创建了一个物品数据集。有关创建数据集的信息，请参阅创建架构和数据集。

单独导入物品（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要将物品导入到的物品数据集的数据集组。
3. 在导航窗格中，选择数据集。
4. 在数据集页面上，选择物品数据集。
5. 在数据集详细信息页面的右上角，选择修改数据集，然后选择创建记录。
6. 在创建物品记录页面中，对于记录输入，以 JSON 格式输入物品详细信息。物品的字段名称和值必须与您在创建物品数据集时使用的架构匹配。Amazon Personalize 提供一个 JSON 模板，其中包含该架构中的字段名称和数据类型。
7. 选择创建记录。在响应中，将列出导入结果并显示成功或失败消息。

单独导入物品 (AWS CLI)

使用 PutItems 操作将一个或多个物品添加到您的物品数据集。通过单次 `PutItems` 调用最多可以导入 10 个物品。此部分假定您已经创建了一个物品数据集。有关创建数据集的信息，请参阅创建架构和数据集。

通过 AWS CLI使用以下 `put-items` 命令添加一个或多个物品。将 `dataset arn` 替换为数据集的 Amazon 资源名称 (ARN)，将 `item Id` 替换为物品 ID。如果具有相同 `itemId` 的物品已存在于物品数据集中，则 Amazon Personalize 会将其替换为新物品。

对于 `properties`，针对物品数据集中的每个字段，将 `propertyName` 替换为架构中的字段名称（采用混合大小写）。例如，GENRES 将是 `genres`，CREATION_TIMESTAMP 将是 creationTimestamp。将 `item data` 替换为物品的数据。`CREATION_TIMESTAMP` 数据必须采用 Unix 纪元时间格式（以秒为单位）。对于分类字符串数据，要包含单个属性的多个类别，请使用竖线 (`|`) 分隔每个类别。例如 `\"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\"}"
    }]'
```

单独导入物品 (AWS SDKs)

使用 PutItems 操作将一个或多个物品添加到您的物品数据集。通过单次 PutItems 调用最多可以导入 10 个物品。如果具有相同 itemId 的物品已存在于物品数据集中，则 Amazon Personalize 会将其替换为新物品。此部分假定您已经创建了一个物品数据集。有关创建数据集的信息，请参阅创建架构和数据集。

以下代码显示如何将一个或多个物品添加到物品数据集。对于每个属性名称参数，传递架构中的字段名称（采用混合大小写）。例如，GENRES 将是 genres，CREATION_TIMESTAMP 将是 creationTimestamp。对于每个属性值参数，传递物品的数据。CREATION_TIMESTAMP 数据必须采用 Unix 纪元时间格式（以秒为单位）。

对于分类字符串数据，要包含单个属性的多个类别，请使用竖线 (|) 分隔每个类别。例如 "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();
```

单独导入操作

在完成创建架构和数据集以创建操作数据集后，您可以单独将一个或多个新操作导入到数据集中。单独导入操作时，随着目录的增长，可以通过小批量导入使操作数据集保持最新状态。您一次最多可以导入 10 个操作。如果您有大量新操作，我们建议您先批量导入数据，然后根据需要单独导入用户操作。请参阅通过数据集导入作业将批量数据导入到 Amazon Personalize。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs 导入操作。如果您导入的操作的 `actionId` 与操作数据集中已有的操作相同，则 Amazon Personalize 会将其替换为新操作。

有关新记录如何影响建议的信息，请参阅训练后更新数据集中的数据。

主题

- 单独导入操作（控制台）
- 单独导入操作 (AWS CLI)
- 单独导入操作 (AWS SDKs)

单独导入操作（控制台）

一次最多可以将 10 个操作导入到操作数据集中。本节假定您已经创建了一个操作数据集。有关创建数据集的信息，请参阅创建架构和数据集。

单独导入操作（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要添加到的操作数据集的数据集组。
3. 在导航窗格中，选择数据集。
4. 在数据集页面上，选择操作数据集。
5. 在数据集详细信息页面的右上角，选择修改数据集，然后选择创建记录。

6. 在创建操作记录页面中，对于记录输入，以 JSON 格式输入操作详细信息。操作的字段名称和值必须与您在创建操作数据集时使用的架构匹配。Amazon Personalize 提供一个 JSON 模板，其中包含该架构中的字段名称和数据类型。
7. 选择创建记录。在响应中，将列出导入结果并显示成功或失败消息。

单独导入操作 (AWS CLI)

使用 PutActions API 操作将一个或多个操作添加到您的操作数据集。您一次最多可以导入 10 个操作。本节假定您已经创建了一个操作数据集。有关创建数据集的信息，请参阅创建架构和数据集。

通过 AWS CLI使用以下 put-actions 命令添加一个或多个操作。将 dataset arn 替换为数据集的 Amazon 资源名称 (ARN)，将 actionId 替换为操作 ID。如果具有相同 actionId 的操作已存在于操作数据集中，则 Amazon Personalize 会将其替换为新操作。

对于 properties，针对操作数据集中的每个字段，将 propertyName 替换为架构中的字段名称（采用驼峰式大小写）。例如，ACTION_EXPIRATION_TIMESTAMP 将是 actionExpirationTimestamp，CREATION_TIMESTAMP 将是 creationTimestamp。将 property data 替换为属性的数据。

```
aws personalize-events put-actions \
  --dataset-arn dataset arn \
  --actions '[{
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    },
    {
      "actionId": "actionId",
      "properties": "{\"propertyName\": "\property data\"}"
    }]'
```

单独导入操作 (AWS SDKs)

使用操作向操作数据集添加一个或多个 PutActions 操作。通过单次 PutActions 调用最多可以导入 10 个操作。如果具有相同 actionId 的操作已存在于操作数据集中，则 Amazon Personalize 会将其替换为新操作。本节假定您已经创建了一个操作数据集。有关创建数据集的信息，请参阅创建架构和数据集。

以下代码显示如何将一个或多个操作添加到您的操作数据集。为每个操作指定 actionId。如果具有相同 actionId 的操作已存在于操作数据集中，则 Amazon Personalize 会将其替换为

新操作。对于 properties，针对操作数据集中的每个其它字段，将 propertyName 替换为架构中的字段名称（采用驼峰式大小写）。例如，ACTION_EXPIRATION_TIMESTAMP 将是 actionExpirationTimestamp，CREATION_TIMESTAMP 将是 creationTimestamp。将 property data 替换为属性的数据。

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

分析 Amazon Personalize 数据集中的数据质量和数量

将数据导入物品交互、用户或物品数据集后，您可以使用 Amazon Personalize 控制台来分析数据。您可以通过数据见解以及列和行统计信息来了解您的数据。您还可以了解可采取哪些操作来改善数据。这些操作可以帮助您满足 Amazon Personalize 资源要求，例如模型训练要求，也可以改进建议。

> **Important**
>
> 您不能使用 Amazon Personalize 控制台来分析操作交互或操作数据集中的数据。

做出任何建议的更改后，您可以再次导入数据，看看是否解决了任何问题或改进了数据集统计信息。有关更新数据的信息，请参阅训练后更新数据集中的数据。

如果未看到任何见解，则说明您的数据符合 Amazon Personalize 数据预期。您可以分析域数据集组或自定义数据集组中的数据。

在生成见解和计算统计信息时，Amazon Personalize 会考虑来自非匿名用户的所有批量和流式传输数据。只有将匿名用户的事件与 `userId` 关联后，才会考虑此类事件。有关更多信息，请参阅 为匿名用户记录事件。

主题

- 分析数据所需的权限
- 数据见解
- 查看数据集见解和统计信息

分析数据所需的权限

如果您授予用户对 Amazon Personalize 的完全访问权限，则无需更改权限。如果您仅向用户授予在 Amazon Personalize 中执行任务所需的权限，则您的 AWS Identity and Access Management (IAM) 策略必须包括以下其他数据洞察操作。

- 个性化：CreateDataInsightsJob
- 个性化：ListDataInsightsJobs
- 个性化：DescribeDataInsightsJob
- 个性化：GetDataInsights

数据见解

以下是您可以在 Amazon Personalize 中生成的可能的数据见解。

见解	操作	相关数据集
交互数据集只有 X 次交互。模型训练至少需要 1000 次交互。我们建议至少 5 万次交互。	在训练模型之前，导入 Y 条额外唯一交互记录。	物品交互
交互数据集只有 X 个具有两个或更多交互的唯一用户。模型训练需要至少 25 个这样的用户。我们建议至少 1000 个用户。	为其他 Y 个用户中的每个用户导入至少 2 条交互记录。	物品交互
物品数据集中有 X% 的物品在交互数据集中没有交互，因此可能不建议使用它们。	请务必导入所有互动数据，并检查您的项目和互动数据集 IDs 之间是否不匹配。查看下面有关物品和交互数据集的数据集统计信息，确保导入的行数达到了预期的行数。如果您的使用案例或食谱使用浏览，请修改浏览配置，以建议更多没有交互数据的物品。	物品交互和物品
用户数据集中有 X% 的用户在交互数据集中没有交互。这些用户将收到热门物品的建议。	请务必导入所有互动数据，并检查用户和互动数据集 IDs 之间是否存在不匹配的情况。查看下面有关用户和交互数据集的数据集统计信息，确保导入的行数达到了预期的行数。导入任何其他交互，以便更多用户拥有交互数据。	物品交互和用户
<用户或物品或交互> 数据集有 X% 的行缺失值。这可能会对建议产生负面影响。我们建议所有必填字段和可选字段的填写率至少为 70%。	导入其他完整记录，或者在没有不完整行的情况下再次导入数据，或者在将缺失值替换为替代数据（例如数值列的平均值或类别列的最常见值）的情况下再次导入数据。	任何
<datasetType>数据集中以下各列的完成度不到 70%：< ColumnName、	导入其他完整记录，或者在没有不完整行的情况下再次导入数据，或者在将缺失值	任何

见解	操作	相关数据集
ColumnName... >。如果这些数据包含在训练中，可能会对建议产生负面影响。我们建议允许空值的列的填写率至少为 70%。	替换为替代数据（例如数值列的平均值或类别列的最常见值）的情况下再次导入数据。	
以下（数字）列具有异常值：< ColumnName、 ColumnName... >。异常值不一定是问题，但有时会对建议产生负面影响。	使用下面的列统计信息，检查这些列的最小值和最大值是否符合您的预期。如果这些值出乎意料，请检查这些列中的数据是否存在不准确之处，并查看您的数据收集和数据处理是否存在问题。	任何
以下列有 1000 多个可能的类别：< ColumnName、 ColumnName... >。如果这些数据包含在训练中，则可能会对建议产生负面影响：<ColumnName，ColumnName... >。	检查您的分类数据是否存在问题，例如拼写差异导致的重复类别。纠正任何不准确之处并重新导入数据。	任何
以下文本元数据列的完成率不到 85%，不会用于模型训练：<ColumnName，ColumnName... >。	导入其他行或使用这些列的文本数据重新导入行。	物品
交互数据集有 10 多个唯一的事件类型，这将导致模型训练失败。	检查您的事件类型列中是否存在不准确之处，例如拼写差异导致的事件类型重复。删除不必要的事件类型并重新导入数据。	物品交互
交互数据集的所有记录都具有相同的时间戳。如果您使用 USER_SEGMENTATION 食谱，并且所有记录都具有相同的时间戳，则模型训练将失败。	检查您的数据是否存在时间戳问题，并将重复的时间戳替换为唯一的时间戳。	物品交互

查看数据集见解和统计信息

要在 Amazon Personalize 数据集中查看您数据的见解和统计信息，请在 Amazon Personalize 控制台中导航到您的数据集，然后选择“运行分析”。

> **Note**
>
> 您必须先导入数据，然后 Amazon Personalize 才能对其进行分析。有关更多信息，请参阅 将训练数据导入 Amazon Personalize 数据集。

查看见解和统计信息

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 从导航窗格中，在数据集下选择数据分析。
4. 在右上角，选择运行分析。此时，Amazon Personalize 开始分析您的数据。这一过程耗时最多 15 分钟。如果成功，结果将显示在此页面上。
5. 在见解中，使用以下内容筛选显示的见解。
   - 要查找包含特定语言的见解，请在查找见解中输入您的标准。在您输入文本时，列表会更新为仅包含在见解或建议操作中包含精确字符串的见解。
   - 要按数据集类型筛选见解，请将所有数据集更改为特定数据集类型。列表将更新为仅包含与此数据集相关的见解。
6. 要查看数据集的数据集统计信息，请执行以下操作。
   - 要查看有关数据集的一般详细信息和统计信息（例如交互数据集中的行数、唯一用户数和唯一物品数），请展开该数据集对应的部分。
   - 要查看某列的详细统计信息，请展开数据集部分，选择列级别统计信息，然后选择该列对应的单选按钮。
7. 更正数据中的所有问题，重新导入，然后运行另一次分析进行验证。有关再次导入数据的更多信息，请参阅训练后更新数据集中的数据。

Amazon Personalize 中的域推荐器

如果创建了域数据集组，则在导入数据后，就可以为域应用场景创建推荐器。推荐器 是生成建议的域数据集组资源。您可以在应用程序中使用推荐器来获取有关 GetRecommendations 操作的实时建议。

创建推荐器时，您需指定一个使用案例，然后 Amazon Personalize 通过使用案例的最佳配置，训练支持推荐器的模型。每个使用案例对获取建议都有不同的 API 要求。有关按域划分的推荐器使用案例列表，请参阅选择使用案例。每个区域最多可以创建 15 个推荐器。

Amazon Personalize 每 7 天自动重新训练一次支持推荐器的模型。这是一次全面的再训练，基于数据集中的全部数据创建全新的模型。Amazon Personalize 通过热门精选 和为您推荐 使用案例，每两小时更新一次现有模型，以便在建议中加入带有浏览的新物品。

创建推荐器时，您可以执行以下操作：

- 在推荐中启用物品元数据。有关更多信息，请参阅 在建议中启用元数据。
- 配置在训练支持推荐器的模型时使用的列。有关更多信息，请参阅 配置在创建 Amazon Personalize 域推荐器时使用的列。
- 对于热门精选和为您推荐应用场景，您可以配置浏览。有关更多信息，请参阅 为域推荐器配置浏览。

创建推荐器后，您可以执行以下操作：

- 评估推荐器 - 您可通过离线和在线指标来评估您的推荐器的性能。有关更多信息，请参阅 评估 Amazon Personalize 域推荐器。
- 停止和重新启动推荐器 - 如果您想暂停为活动推荐器计费，可以停止推荐器并稍后重新启动。有关更多信息，请参阅 停止推荐器。
- 更新推荐器的配置 - 您可以更新推荐器在训练中使用的列并更新推荐器的请求容量。有关更多信息，请参阅 更新推荐器。
- 删除推荐器 - 您可以通过 DeleteRecommender 操作删除推荐器。或者，您可以从 Amazon Personalize 控制台的推荐器详情页面中删除推荐器。

主题

- 推荐器状态
- 每秒最低建议请求数和自动扩缩
- 在 Amazon Personalize 中创建域推荐器

- 在推荐中为 Amazon Personalize 中的域推荐器启用元数据
- 配置在创建 Amazon Personalize 域推荐器时使用的列
- 为域推荐器配置浏览
- 评估 Amazon Personalize 域推荐器
- 更新推荐器
- 停止推荐器

推荐器状态

推荐器可处于以下一种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

要获得推荐器状态，请导航至 Amazon Personalize 控制台中的“推荐器”页面或使用 DescribeRecommender 操作。

每秒最低建议请求数和自动扩缩

> **Important**
>
> 高 minRecommendationRequestsPerSecond 会增加您的账单费用。我们建议从 1 开始设置 minRecommendationRequestsPerSecond（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要minRecommendationRequestsPerSecond时增加使用量。

创建推荐器时，您可以配置推荐器的每秒最低建议请求数。每秒最低建议请求数 (minRecommendationRequestsPerSecond) 指定了 Amazon Personalize 预调配的基准建议请求吞吐量。默认值 minRecommendationRequestsPerSecond 为1。建议请求是单个 GetRecommendations 操作。请求吞吐量以每秒请求数来衡量，Amazon Personalize 使用每秒请求数来计算每小时请求数和推荐器使用量的价格。

如果您的每秒请求数增加到超过 minRecommendationRequestsPerSecond，则 Amazon Personalize 会自动纵向扩展和缩减预调配容量，但永远不会低于 minRecommendationRequestsPerSecond。增加容量时会有短时间延迟，这可能会导致请求丢失。

您的账单是每小时最低请求数（基于 minRecommendationRequestsPerSecond）或实际请求数中的较大值。使用的实际请求吞吐量按一小时窗口 requests/second 内的平均值计算。我们建议从默认值开始minRecommendationRequestsPerSecond，使用 Amazon CloudWatch 指标跟踪您的使用情况，然后minRecommendationRequestsPerSecond根据需要增加使用量。

在 Amazon Personalize 中创建域推荐器

您可以使用 Amazon Personalize 控制台创建推荐人， AWS Command Line Interface (AWS CLI) 或 AWS SDKs。下面包括使用 Amazon Personalize 控制台创建推荐器的详细步骤，还包括了代码示例，用于演示如何创建仅包含必需字段的推荐器。

- 有关如何在推荐中启用元数据的代码示例，请参阅在建议中启用元数据。
- 有关如何配置在训练支持推荐器的模型时使用的列的代码示例，请参阅配置在创建 Amazon Personalize 域推荐器时使用的列。
- 有关如何为 `Top picks for your` 或 `Recommended for you` 应用场景配置浏览的代码示例，请参阅为域推荐器配置浏览。

主题

- 创建推荐器（控制台）
- 创建推荐器（AWS CLI）
- 创建推荐器（AWS SDKs）

创建推荐器（控制台）

> **Important**
>
> 高 minRecommendationRequestsPerSecond 会增加您的账单费用。我们建议从 1 开始设置 minRecommendationRequestsPerSecond（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要minRecommendationRequestsPerSecond时增加使用量。有关更多信息，请参阅 每秒最低建议请求数和自动扩缩。

使用 Amazon Personalize 控制台为您的每个使用案例创建推荐器，如下所示。如果您刚刚创建了域数据集组，且已经进入了概述页面，请跳至步骤 3。

创建推荐器

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择您的域数据集组。
3. 在步骤 3 中，选择使用 <域名> 推荐器，然后选择创建推荐器。
4. 在选择使用案例页面上，选择要创建推荐器的使用案例，并分别提供推荐器名称。Amazon Personalize 会为您选择的每个使用案例创建一个推荐器。可用的使用案例取决于您的域。有关选择使用案例的信息，请参阅选择使用案例。
5. 选择下一步。
6. 在高级配置页面上，根据您的业务需求配置每个推荐器：

   - 对于推荐器使用案例使用的每个数据集，您可以选择 Amazon Personalize 在训练支持推荐器的模型时考虑的列。默认情况下，Amazon Personalize 使用可在训练时采用的所有列。有关更多信息，请参阅 配置在创建 Amazon Personalize 域推荐器时使用的列。
   - 您可以修改每秒最低建议请求数，为推荐器指定新的最低请求容量。高 `minRecommendationRequestsPerSecond` 会增加您的账单费用。我们建议从 1 开始设置（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要`minRecommendationRequestsPerSecond`时增加使用量。有关更多信息，请参阅 每秒最低建议请求数和自动扩缩。
   - 如果您希望能够在建议中包含物品数据集元数据，请选择在建议结果中返回物品元数据。如果启用，则可以在建议或个性化排名请求中指定物品数据集中的列。Amazon Personalize 会在建议响应中返回每个物品的这些数据。

     要启用元数据，必须拥有包含一列元数据的物品数据集。
   - 对于 `Top picks for your` 或 `Recommended for you` 使用案例，可以选择更改浏览配置。浏览包括测试不同的物品建议，以了解用户如何对交互数据很少的物品做出响应。使用以下字段来配置浏览：
     - 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
     - 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品

使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

- 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。

7. 要为您的每个使用案例创建推荐器，请选择创建推荐器。

   您可以在推荐器页面上，监控每个推荐器的状态。当推荐器状态为“活动”时，您可以在应用程序中使用它来获取建议。

创建推荐器（AWS CLI）

使用以下 AWS CLI 代码为域名用例创建推荐器。针对您的每个域使用案例运行此代码。对于 `recipeArn`，为您的使用案例提供 Amazon 资源名称 (ARN)。可用的使用案例取决于您的域。有关用例及其列表， ARNs 请参阅选择使用案例。

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN
```

创建推荐器（AWS SDKs）

使用以下代码为域使用案例创建推荐器。为推荐器命名，并提供域数据集组的 Amazon 资源名称 (ARN)。对于 `recipeArn`，为您的使用案例提供 ARN。针对您的每个域使用案例运行此代码。可用的使用案例取决于您的域。有关用例、用例及其 ARNs要求的列表，请参阅选择使用案例。

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();
```

在推荐中为 Amazon Personalize 中的域推荐器启用元数据

> **Important**
>
> 在建议中启用元数据时，会产生额外成本。有关更多信息，请参阅 Amazon Personalize 定价。

创建推荐器时，您可以启用该选项，以将物品数据集中的物品元数据与建议结果一起包括在内。如果启用，则可以在建议请求中指定物品数据集中的列。Amazon Personalize 会在建议响应中返回每个物品的这些数据。

您可以使用元数据来丰富用户界面中的建议，例如将电影的类型添加到轮播中。或者，您可以用它来直观地评测建议质量。如果您在应用程序中使用生成式人工智能，则可以将元数据插入人工智能提示中，以生成相关性更高的内容。有关 Amazon Personalize 与生成式人工智能结合使用的更多信息，请参阅Amazon Personalize 和生成式人工智能。

要向建议添加元数据，您必须拥有包含一列元数据的物品数据集。您不一定要在训练中使用元数据。有关创建数据集的信息，请参阅创建架构和数据集。有关在数据集中更新数据的信息，请参阅训练后更新数据集中的数据。

以下代码示例演示如何启用选项以在 AWS CLI 或中包含项目元数据 AWS SDKs。要使用 Amazon Personalize 控制台执行此操作，请在创建推荐器时在高级配置页面上启用元数据。有关更多信息，请参阅 创建推荐器（控制台）。

启用元数据（AWS CLI）

如果您有物品数据集，并且希望该选项在您获得建议时包含元数据，请在 `recommender-config` 中将 `enableMetadataWithRecommendations` 设置为 `true`。

```
aws personalize create-recommender \
```

```
--name recommender name \
--dataset-group-arn dataset group \
--recipe-arn recipe ARN \
--recommender-config "{\"enableMetadataWithRecommendations\": "true"}"
```

启用元数据 (AWS SDK)

如果您有物品数据集，并且希望该选项在您获得建议时包含元数据，请在 recommender-config 中将 enableMetadataWithRecommendations 设置为 true。

```
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)
```

配置在创建 Amazon Personalize 域推荐器时使用的列

在创建推荐器时，您可以修改 Amazon Personalize 在训练支持推荐器的模型时考虑的列。

您可以这样做，以试验不同的训练数据组合；也可以排除无有意义数据的列。例如，可能有一列，您只想将其用来筛选建议。您可以将此列排除在训练之外，Amazon Personalize 只有在筛选时才会考虑该列。

您无法排除 EVENT_TYPE 列。默认情况下，Amazon Personalize 使用可在训练时采用的所有列。始终从训练中排除以下数据：

- 具有布尔数据类型的列
- 展示数据
- 既不是分类也不是文本的自定义字符串字段

您不能在训练中包括展示数据，但如果您的应用场景或配方使用了这些数据，Amazon Personalize 会在您获取建议时使用展示数据来指导浏览。

以下代码示例显示了使用 AWS CLI 或进行训练时如何配置所使用的列 AWS SDKs。要使用 Amazon Personalize 控制台执行此操作，请在创建推荐器时在高级配置页面上指定要使用的列。有关更多信息，请参阅 创建推荐器（控制台）。

配置训练时使用的列 (AWS CLI)

要将列排除在训练之外，请在推荐器配置期间，在 `trainingDataConfig` 中提供 `excludedDatasetColumns` 对象。对于对象中的每个键，提供数据集类型。对于每个值，提供要排除的列的列表。有关更多信息，请参阅 配置在创建 Amazon Personalize 域推荐器时使用的列。

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

配置训练时使用的列 (AWS SDKs)

要将列排除在训练之外，请在推荐器配置期间，在 `trainingDataConfig` 中提供 `excludedDatasetColumns` 对象。对于每个键，提供数据集类型。对于每个值，提供要排除的列的列表。以下代码显示了如何在创建推荐器时将列排除在训练之外。有关更多信息，请参阅 配置在创建 Amazon Personalize 域推荐器时使用的列。

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();
```

为域推荐器配置浏览

对于 Top picks for your 或 Recommended for you 使用案例，Amazon Personalize 在推荐物品时使用浏览。浏览包括测试不同的物品建议，以了解用户如何对交互数据很少的物品做出响应。您可以使用以下方式配置浏览：

- 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
- 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

  要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

以下代码示例展示了如何使用 AWS CLI 或为推荐器配置探索。 AWS SDKs要使用 Amazon Personalize 控制台执行此操作，请在创建推荐器时在高级配置页面上指定浏览配置。有关更多信息，请参阅 创建推荐器（控制台）。

配置浏览（AWS CLI）

以下代码显示了如何在为 Top picks for you 使用案例创建推荐器时配置浏览。本示例使用了默认值。

如果您有物品数据集，并且希望该选项在您获得建议时包含元数据，请更新 recommender-config 以添加 enableMetadataWithRecommendations 字段并将其设置为 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\"}}"
```

配置浏览（AWS SDKs）

对于 Top picks for your 或 Recommended for you 使用案例，Amazon Personalize 在推荐物品时使用浏览。浏览包括测试不同的物品建议，以了解用户如何对交互数据很少的物品做出响应。您可以使用以下方式配置浏览：

- 重点浏览不太相关的物品（浏览权重）- 配置要浏览的程度。指定 0 到 1 之间的十进制值。默认值为 0.3。值越接近 1，浏览次数就越多。浏览次数越多，建议包括的物品就越多，这些物品具有的物品交互数据更少或基于先前行为的相关性更低。如果值为零，则表示没有浏览，建议基于当前数据（相关性）。
- 浏览物品使用期限截止时间 - 指定自物品交互数据集中所有物品最近一次交互以来的最大物品使用期限（以天为单位）。这会根据物品使用期限定义物品浏览的范围。Amazon Personalize 根据物品的创建时间戳或物品交互数据（如果缺少创建时间戳数据）来确定物品使用期限。有关 Amazon Personalize 如何确定物品使用期限的更多信息，请参阅创建时间戳数据。

  要增加 Amazon Personalize 在浏览期间考虑的物品数量，请输入更大的值。最小值为 1 天，默认值为 30 天。建议可能包括比您指定的物品使用期限截止时间更早的物品。这是因为这些物品与用户相关，而浏览并未识别出它们。

以下代码显示了如何在创建推荐器时配置浏览。本示例使用了默认值。

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();
```

评估 Amazon Personalize 域推荐器

您可通过离线和在线指标来评估您的推荐器的性能。在线指标 是您在用户与实时建议的交互中观察到的实证结果。例如，您可以记录用户在浏览您的目录时的点击率。您负责生成和记录任何在线指标。

离线指标 是 Amazon Personalize 在您创建推荐器时生成的指标。您可使用离线指标来评估推荐器底层模型的性能。您可以查看修改推荐器配置产生的影响，并且可以比较同一数据集组中使用相同数据但针对不同应用场景进行训练的推荐器的结果。

避免比较使用不同数据训练的不同推荐器的指标。指标的差异可能源于数据的差异，而不是源于模型性能。例如，您可能有一个数据集组，其中包含每个用户的稀疏 `purchase` 事件数据，而另一个数据集组则包含可靠的 `view` 事件数据。根据像 `precision at K` 这样的指标，由于交互次数较多，根据查看事件数据训练的推荐器可能会错误地表现得更好。

为了获得性能指标，Amazon Personalize 将输入的交互数据拆分为训练集和测试集。训练集由 90% 的用户及其交互数据组成。测试集由剩余的 10% 的用户及其交互数据组成。

然后，Amazon Personalize 使用训练集创建推荐器。训练完成后，Amazon Personalize 会向新推荐器提供测试集中每位用户 90% 的最早数据作为输入。然后，Amazon Personalize 通过将推荐器生成的建议与测试集中每位用户 10% 的最新数据中的实际交互进行比较，计算指标。

主题

- 检索指标
- 指标定义
- 示例
- 其他资源

检索指标

在您的推荐器处于活动状态后，您可以在 Amazon Personalize 控制台中查看推荐器的指标，或者通过调用 DescribeRecommender 操作来检索指标。

主题

- 查看指标（控制台）
- 检索指标 (AWS CLI)
- 检索指标 (AWS 软件开发工具包)

查看指标（控制台）

要在控制台中查看推荐器指标，请导航到推荐器的详细信息页面。

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 在数据集组页面上，选择您的域数据集组。
3. 在导航窗格中，选择推荐器。
4. 从推荐器列表中，选择要查看其指标的那个推荐器。

检索指标 (AWS CLI)

以下代码显示了如何使用 AWS CLI获取推荐器的指标。

```
aws personalize describe-recommender \
--recommender-arn recommender arn
```

以下是为 VIDEO_ON_DEMAND 域的热门精选 使用案例创建的推荐器指标输出的示例。

```
{
    "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",
    }
}
```

检索指标 (AWS 软件开发工具包)

以下代码显示了如何使用 适用于 Python (Boto3) 的 AWS SDK获取推荐器的指标。

```
import boto3

personalize = boto3.client('personalize')

response = personalize.describe_recommender(
    recommenderArn = 'recommender_arn'
)
print(response['recommender']['modelMetrics'])
```

以下是为 VIDEO_ON_DEMAND 域的热门精选 使用案例创建的推荐器指标输出的示例。

```
{
    "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",
    }
}
```

指标定义

下文使用以下术语描述了 Amazon Personalize 为推荐器生成的指标：

- 相关建议 是针对用户实际与之交互的物品的建议。这些物品来自测试集中每个用户 10% 的最新交互数据。
- 排名 是指推荐项在推荐列表中的位置。位置 1（列表的顶部）被假定为与用户最相关。

对于每个指标，数字越大（越接近 1）越好。要深入了解相关信息，请参阅其他资源中列出的资源。

覆盖

覆盖率 的值可指明，Amazon Personalize 可能推荐的唯一物品占交互数据集和物品数据集中唯一物品总数的比例。覆盖率分数越高，意味着 Amazon Personalize 推荐的物品越多，而不是向不同用户重复推荐相同的几件物品。针对具有物品浏览功能的使用案例（例如热门精选 (VIDEO_ON_DEMAND) 和为您推荐 (ECOMMERCE)），其覆盖率比不具备这一功能的使用案例更高。

前 25 项中平均倒数排名

该指标可指明，模型在排名靠前的位置生成相关建议的能力。如果您要为用户生成相关搜索结果，且不希望用户选择列表中排名靠后的物品，则可以选择前 25 项中平均倒数排名 较高的模型。例如，用户经常在搜索结果中选择第一个烹饪食谱。

Amazon Personalize 使用建议请求的平均倒数排名分数来计算该指标。每个倒数排名分数的计算方法如下：`1 / the rank of the highest item interacted with by the user`，其中可能的总排名为 25。用户与之交互的其他排名较低的物品将被忽略。如果用户选择了第一个物品，则分数为 1。如果没有选择任何物品，则分数为 0。

例如，您可以向三个不同的用户分别显示 25 条建议：

- 如果用户 1 点击等级 4 的物品和等级 10 的物品，则他们的倒数等级分数为1/4。
- 如果用户 2 点击等级 2 的物品、等级 4 的物品和等级 12 的物品，则他们的倒数等级分数为 1/2。
- 如果用户 3 点击了等级 6 的单件物品，则他们的倒数等级分数为 1/6。

所有建议请求的平均倒数排名（在本例中为 3）的计算公式为 `(1/4 + 1/2 + 1/6) / 3 = .3056`。

K（5、10 或 25）项标准化折扣累计增益 (NDCG)

该指标可指明模型对建议的排名，其中 K 是样本大小，即 5、10 或 25 条建议。如果您最感兴趣的是建议的排名，而不仅仅是排名最高的物品，则此指标非常有用（有关此信息，请参阅 `mean reciprocal rank at 25`）。例如，如果您的应用程序在轮播中一次最多显示 10 部电影，则 `NDCG at 10` 分数将非常有用。

Amazon Personalize 根据测试集中每位用户的排名位置为建议分配权重，来计算 NDCG。每条建议被折扣一倍（给定较低的权重），具体取决于其位置。最终指标是测试集中所有用户的平均值。K 项标准化折扣累计增益假定列表中排名较低的推荐与列表中排名较高的推荐相比相关性更低。

Amazon Personalize 使用加权系数 `1/log(1 + position)`，其中列表的顶部是位置 1。

K 项精度

该指标可指明，根据样本大小 K（5、10 或 25）条建议，模型建议的相关性。

Amazon Personalize 按照以下方法计算此指标：测试集中每个用户前 K 条建议中的相关建议数量除以 K，其中 K 为 5、10 或 25。最终指标是测试集中所有用户的平均值。

例如，如果您向用户推荐 10 个物品，而用户与其中 3 个物品交互，则 K 项精度为：3 个正确预测的物品除以总计 10 个建议物品 (`3 / 10 = .30`)。

此指标奖励相关物品的精准建议。分数越接近 1，模型就越精确。

示例

以下是可为特定用户生成建议列表的推荐器的简单示例。第二个和第五个推荐与此用户的测试数据中的记录匹配。这些是相关的推荐。如果将 K 设置为 5，将为该用户生成以下指标。

reciprocal_rank

计算： 1/2

结果：0.5000

normalized_discounted_cumulative_gain_at_5

计算：(1/log(1 + 2) + 1/log (1 + 5))/(1/log(1 + 1) + 1/log (1 + 2))

结果：0.6241

precision_at_5

计算： 2/5

结果：0.4000

其他资源

要深入了解推荐器系统的不同类型的指标，请参阅以下外部资源：

- MRR vs MAP vs NDCG： Rank-Aware 评估指标以及何时使用它们
- 折扣累计增益：您应该了解的排名指标
- 推荐器系统的 K 项查全率和精度
- 推荐器系统的排名评估指标

更新推荐器

创建推荐器后，您可以更新推荐器的配置：

- 您可以更新推荐器在训练中使用的列。如果您修改训练时使用的列，则 Amazon Personalize 会自动开始对支持推荐器的模型进行全面的重新训练。更新完成后，您仍然可以从推荐器中获得建议。在更新完成之前，推荐器使用之前的配置。要跟踪此更新的状态，请使用 DescribeRecommender 操作中返回的 `latestRecommenderUpdate`。如果您提供的列与创建推荐器时提供的列相同，则不会进行更新。
- 您可以更新推荐器的最低每秒建议请求数。这指定了 Amazon Personalize 预调配的基准建议请求吞吐量。高值会增加您的账单费用。我们建议从 1 开始。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加该值。有关更多信息，请参阅 每秒最低建议请求数和自动扩缩。
- 对于热门精选 和为您推荐 使用案例，您可以通过调整浏览相关物品的重视程度和浏览物品期限截止时间，更新浏览配置。有关浏览的信息，请参阅选择使用案例中与您的使用案例对应的部分。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 更新推荐器。

主题

- 更新推荐器（Amazon Personalize 控制台）
- 更新推荐器 (AWS CLI)

- 更新推荐器 (AWS SDK)

更新推荐器（Amazon Personalize 控制台）

创建推荐器后，您可以对其更新。您可以更新推荐器在训练中使用的列以及推荐器的最低每秒建议请求数。对于热门精选 和为您推荐 使用案例，您可以更新浏览配置。要使用控制台更新推荐器，请执行以下操作。

更新推荐器的配置（控制台）

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 在数据集组页面上，选择您的域数据集组。
3. 在导航窗格中，选择推荐器。
4. 在推荐器页面上，选择要更新的推荐器。
5. 在推荐器配置中，选择编辑。
6. 更改推荐器的配置，然后选择更新。有关不同配置选项的信息，请参阅创建推荐器（控制台）。

更新推荐器 (AWS CLI)

要通过 AWS CLI 更新推荐器，请使用 `update-recommender` 命令。提供推荐器的 Amazon 资源名称 (ARN) 以及更新的配置。以下代码显示如何更新推荐器用于训练的列。

```
aws personalize update-recommender \
--dataset-group-arn dataset group ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

如果您修改训练中使用的列，则 Amazon Personalize 会自动开始对支持推荐器的模型进行全面的重新训练。更新完成后，您仍然可以从推荐器中获得建议。在更新完成之前，推荐器使用之前的配置。要跟踪此更新的状态，请使用 DescribeRecommender 操作中返回的 `latestRecommenderUpdate`。

有关您可以更改的不同配置的更多信息，请参阅 RecommenderConfig。

更新推荐器 (AWS SDK)

要通过 AWS 更新推荐器，请使用 UpdateRecommender 操作。提供推荐器的 Amazon 资源名称 (ARN)，并指定新配置。以下代码显示如何更新推荐器用于训练的列。

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();
```

如果您在 `recommenderConfig` 的 `excludedDatasetColumns` 中修改训练中使用的列，则 Amazon Personalize 会自动开始对支持推荐器的模型进行全面的重新训练。更新完成后，您仍然可以从推荐器中获得建议。在更新完成之前，推荐器使用之前的配置。要跟踪此更新的状态，请使用 DescribeRecommender 操作中返回的 `latestRecommenderUpdate`。

有关您可以更改的不同配置的更多信息，请参阅 RecommenderConfig。

停止推荐器

在您的推荐器处于活动状态后，您可以停止推荐器并稍后再启动。这样，您就可以暂停推荐器计费，并仅在使用推荐器时才付费。例如，您可能只需要在一周中的某几天获得建议。您可以在不需要时停止推荐器，然后在需要时启动推荐器。

停止推荐器后，您无法使用它来获取建议。停止推荐器会停止对推荐器的计费和再训练。但是，停止推荐器不会删除推荐器。您可以随时重启它并恢复获取建议。启动推荐器不会使用您的数据创建新的推荐器。相反，它每 7 天恢复对推荐器的计费和再训练。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface（AWS CLI）、AWS SDK 来停止和启动推荐器。

推荐器状态

停止推荐器时，推荐器状态会按以下顺序从 ACTIVE 变为 INACTIVE：

ACTIVE > STOP PENDING > STOP IN PROGRESS > INACTIVE

启动推荐器时，推荐器状态会按以下顺序从 INACTIVE 变为 ACTIVE：

INACTIVE > START PENDING > START IN PROGRESS > ACTIVE

主题

- 停止推荐器（控制台）
- 停止推荐器（AWS CLI）
- 停止推荐器 (AWS SDK)

停止推荐器（控制台）

您可以从 Amazon Personalize 控制台的推荐器详情页面中停止推荐器。

停止推荐器

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 在数据集组页面上，选择您的域数据集组。
3. 在导航窗格中，选择推荐器。
4. 在推荐器页面上，选择要停止的推荐器。
5. 在推荐器的详情页面上，选择右上角的停止推荐器，然后在显示的窗口中进行确认。当推荐器状态为非活动时，表示推荐器已停止。您可以从这个相同的页面上重新启动推荐器。

停止推荐器（AWS CLI）

要通过 AWS CLI 停止活动推荐器，请使用 `stop-recommender` 命令（使用了 StopRecommender API 操作）并为推荐器提供 Amazon 资源名称（ARN）。要重新启动推荐器，可以使用 `start-recommender` 命令，该命令使用 StartRecommender。以下代码显示如何停止推荐器：

```
aws personalize stop-recommender --recommender-arn "recommender arn"
```

停止推荐器 (AWS SDK)

要通过 AWS SDK 停止活动推荐器，请使用 StopRecommender API 操作，并为推荐器提供 Amazon 资源名称（ARN）。要重新启动它，可以使用 StartRecommender。以下代码显示如何停止推荐器：

SDK for Python (Boto3)

要通过 SDK for Python (Boto3) 停止活动推荐器，请使用 `stop_recommender` 方法并为推荐器提供 Amazon 资源名称 (ARN)，如下所示。

```
import boto3
```

```
personalize = boto3.client('personalize')

stop_recommender_response = personalize.stop_recommender(
    recommenderArn = "recommenderARN"
)
print(stop_recommender_response)
```

SDK for Java 2.x

要通过 SDK for Java 2.x 停止活动推荐器，请使用 stopRecommender 方法并为推荐器提供 ARN，如下所示。

```
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();
```

用于训练和部署 Amazon Personalize 模型的自定义资源

如果您使用的是自定义配方，则在导入数据后，就可以创建用于训练和部署 Amazon Personalize 模型的自定义资源。使用这些资源获取建议。创建自定义资源时，执行以下操作：

1. 创建和配置解决方案：自定义解决方案参数和食谱特定的超参数，使模型满足您的特定业务需求。默认情况下，新的解决方案版本使用自动训练以可配置的频率创建解决方案版本。默认频率为每 7 天一次。有关配置解决方案的更多信息，请参阅配置解决方案。有关 Amazon Personalize 中的自定义配方的更多信息，请参阅选择食谱。
2. 创建解决方案版本（对于不使用自动训练的解决方案）：对于使用自动训练的解决方案，在解决方案处于活动状态后解决方案版本创建会自动开始。对于使用手动训练的解决方案，手动创建解决方案版本。解决方案版本生成 Amazon Personalize 建议或用户细分。有关手动创建解决方案版本的更多信息，请参阅手动创建解决方案版本。要停止创建解决方案版本，请导航至解决方案版本详细信息页面，然后选择停止。有关更多信息，请参阅 停止创建解决方案版本。
3. 评估解决方案版本 - 使用 Amazon Personalize 从新解决方案版本中生成的指标来评估模型的性能。请参阅通过指标评估 Amazon Personalize 解决方案版本。
4. 使用市场活动部署解决方案版本（仅用于实时建议）：创建市场活动以部署您的解决方案版本。您可在请求实时建议时使用市场活动。如果您收到批量建议，则无需创建市场活动。有关更多信息，请参阅 通过市场活动部署 Amazon Personalize 解决方案版本。如果您想要更改现有市场活动的设置，例如在推荐中启用元数据，则必须更新您的市场活动。有关更多信息，请参阅 更新 Amazon Personalize 市场活动的配置。

主题

- 在 Amazon Personalize 中配置自定义解决方案
- 更新解决方案以更改其自动训练配置
- 手动创建解决方案版本
- 停止创建解决方案版本
- 通过指标评估 Amazon Personalize 解决方案版本
- 通过市场活动部署 Amazon Personalize 解决方案版本
- 更新 Amazon Personalize 市场活动的配置

在 Amazon Personalize 中配置自定义解决方案

完成数据导入后，就可以创建解决方案了。解决方案 是指 Amazon Personalize 配方、自定义训练参数和一个或多个解决方案版本的组合。解决方案版本 是指经过训练的机器学习模型。

默认情况下，所有新解决方案都使用自动训练，每 7 天创建一个新的解决方案版本。只有在自上次训练以来导入了批量交互数据或实时交互数据时，才会进行自动训练。这包括物品交互，或者对于使用 Next-Best-Action 配方的解决方案，包括操作交互数据。自动训练会持续运行，直到您删除解决方案。有关更多信息，请参阅 配置自动训练。

如果您有现有解决方案，则可以使用 Amazon Personalize 控制台来克隆解决方案。克隆解决方案时，您可以使用现有解决方案的配置（例如配方和超参数）作为起点，并进行任何更改。有关更多信息，请参阅 克隆解决方案（控制台）。

您可以使用控制台、AWS Command Line Interface（AWS CLI）或 AWS SDK 创建和配置解决方案。创建解决方案后，您可以在 Amazon Personalize 控制台的解决方案详情页面或通过 DescribeSolution 操作，查看解决方案的配置详细信息。

默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

主题

- 创建解决方案
- 配置自动训练
- 配置训练时使用的列
- 针对其他目标优化解决方案
- 使用事件配置优化解决方案
- 超级参数和 HPO
- 选择用于训练的物品交互数据
- 克隆解决方案（控制台）

创建解决方案

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface（AWS CLI）或 AWS SDK 创建自定义解决方案。下面包括使用 Amazon Personalize 控制台创建解决方案的详细步骤，还包括了代码示例，用于演示如何创建仅包含必需字段的解决方案。

主题

- 创建解决方案（控制台）
- 创建解决方案 (AWS CLI)
- 创建解决方案 (AWS SDK)

创建解决方案（控制台）

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

要在控制台中创建解决方案，请选择您的数据集组，然后指定解决方案名称、配方和可选的训练配置。

配置解决方案（控制台）

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 在概览页面上，对于步骤 3，执行以下任一操作：
   - 如果您创建了域数据集组，请选择使用自定义资源，然后选择创建解决方案。
   - 如果您创建了自定义数据集组，则选择创建解决方案。
4. 对于解决方案名称，为您的解决方案指定名称。
5. 对于解决方案类型，选择要创建的解决方案类型。您选择的类型决定了可用的配方。
   - 选择物品建议，以便为用户获取物品建议。例如，个性化的电影建议。
   - 选择操作建议，以便为用户获取操作建议。例如，为用户生成下一个最佳操作，例如下载您的应用程序。
   - 选择用户细分，以根据您的物品数据获取用户细分（用户组）。
6. 对于食谱，选择一个食谱（请参阅选择食谱）。
7. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。

8. 选择下一步。
9. 在训练配置页面上，自定义解决方案以满足您的业务需求。

   - 在自动训练中，选择解决方案是否使用自动训练。如果使用了自动训练，则可以更改 `Automatic training frequency`。默认训练频率为每 7 天一次。

     建议使用自动训练。使用自动训练后，您就可以更轻松地维护推荐相关性。训练频率取决于业务需求、所使用的配方以及导入数据的频率。有关更多信息，请参阅 配置自动训练。有关维护相关性的信息，请参阅维护建议相关性。
   - 在超参数配置中，根据您的配方和业务需求配置任何超参数选项。使用不同超级参数的不同食谱。有关可用的超参数，请参阅选择食谱中的各个配方。
   - 在用于训练的列中，如果您的配方生成了物品推荐或用户细分，则可以选择在创建解决方案版本时 Amazon Personalize 考虑的列。有关更多信息，请参阅 配置训练时使用的列。
   - 在事件配置中，如果您的项目交互数据集包含 EVENT_TYPE 或同时包含 EVENT_TYPE 和 EVENT_VALUE 列，则可以选择使用事件类型和事件值阈值字段，来选择 Amazon Personalize 在训练模型时使用的项目交互数据。有关更多信息，请参阅 选择用于训练的物品交互数据。

     如果您有多个事件类型，并且使用了 User-Personalization-v2 配方或 Personalized-Ranking-v2 配方，则还可以为不同的类型指定不同的权重。例如，您可以配置一个解决方案，为购买事件赋予高于点击事件的权重。有关更多信息，请参阅 使用事件配置优化解决方案。
   - 如果您使用 User-Personalization 食谱或 Personalized-Ranking 食谱食谱，则可以选择指定目标并选择目标灵敏度，以便在相关性之外针对目标优化您的解决方案。目标敏感度配置了 Amazon Personalize 如何在基于目标推荐物品与根据交互数据确定的相关性来推荐物品之间取得平衡。有关更多信息，请参阅 针对其他目标优化解决方案。
10. 选择下一步，并查看解决方案详细信息。创建解决方案后，就不能更改其配置。
11. 选择创建解决方案。创建解决方案后，Amazon Personalize 将在一小时内开始创建第一个解决方案版本。训练开始后，可以在解决方案详细信息页面的解决方案版本部分监控训练。自动创建的解决方案版本的训练类型为“自动”。

    当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

    - 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
    - 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

创建解决方案 (AWS CLI)

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

要使用 AWS CLI 创建解决方案，请使用 `create-solution` 命令。此命令使用 CreateSolution API 操作。以下代码展示了如何创建使用自动训练的解决方案。该解决方案每五天自动创建一个新的解决方案版本。

要使用该代码，请更新代码来为解决方案取名，指定您的数据集组的 Amazon 资源名称（ARN），可以选择更改训练频率，并指定要使用的配方的 ARN。有关配方的信息，请参阅选择食谱。

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

- 我们建议您使用自动训练。使用自动训练后，您就可以更轻松地维护和改进推荐相关性。默认情况下，所有新解决方案都使用自动训练。默认训练频率为每 7 天一次。训练频率取决于业务需求、所使用的配方以及导入数据的频率。有关更多信息，请参阅 配置自动训练。
- 根据您的配方，您可以修改代码来配置配方特定的属性和超参数（请参阅超级参数和 HPO），配置用于训练的列（请参阅配置训练时使用的列 (AWS CLI)），或者筛选用于训练的物品交互数据（请参阅选择用于训练的物品交互数据）。
- 如果您使用 User-Personalization 食谱或 Personalized-Ranking 食谱配方，则除了相关性之外，还可以针对目标优化解决方案。有关更多信息，请参阅 针对其他目标优化解决方案。

创建解决方案后，记录解决方案 ARN 以备将来使用。通过自动训练，解决方案处于活动状态后，解决方案版本的创建将在一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。训练开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的 Amazon 资源名称（ARN）。要获取其状态，请使用 DescribeSolutionVersion API 操作。

当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

- 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
- 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

创建解决方案 (AWS SDK)

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

要使用 AWS SDK 创建解决方案，请使用 CreateSolution API 操作。以下代码展示了如何创建使用自动训练的解决方案。该解决方案每五天自动创建一个新的解决方案版本。

要使用该代码，请更新代码来为解决方案取名，指定您的数据集组的 Amazon 资源名称（ARN），可以选择更改训练频率，并指定要使用的配方的 ARN。有关配方的信息，请参阅选择食谱。

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();
```

- 我们建议您使用自动训练。使用自动训练后，您就可以更轻松地维护和改进推荐相关性。默认情况下，所有新解决方案都使用自动训练。默认训练频率为每 7 天一次。训练频率取决于业务需求、所使用的配方以及导入数据的频率。有关更多信息，请参阅 配置自动训练。
- 根据您的配方，您可以修改代码来配置配方特定的属性和超参数（请参阅超级参数和 HPO），配置用于训练的列（请参阅配置训练时使用的列 (AWS 软件开发工具包)），或者筛选用于训练的物品交互数据（请参阅选择用于训练的物品交互数据）。
- 如果您使用 User-Personalization 食谱或 Personalized-Ranking 食谱配方，则除了相关性之外，还可以针对目标优化解决方案。有关更多信息，请参阅 针对其他目标优化解决方案。

创建解决方案后，记录解决方案 ARN 以备将来使用。通过自动训练，解决方案处于活动状态后，解决方案版本的创建将在一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。训练开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的 Amazon 资源名称（ARN）。要获取其状态，请使用 DescribeSolutionVersion API 操作。

您可以使用以下 Python 代码来等待自动训练开始。`wait_for_training_to_start` 方法返回第一个解决方案版本的 ARN。

```
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)
```

当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

- 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
- 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

配置自动训练

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

创建解决方案时，您可以配置解决方案是否使用自动训练。还可以配置训练频率。例如，您可以将解决方案配置为每五天创建一个新的解决方案版本。

默认情况下，所有新解决方案都使用自动训练，每 7 天创建一个新的解决方案版本。只有在自上次训练以来导入了批量交互数据或实时交互数据时，才会进行自动训练。这包括物品交互，或者对于使用 Next-Best-Action 配方的解决方案，包括操作交互数据。自动训练会持续运行，直到您删除解决方案。

我们建议您使用自动训练。自动训练使维护解决方案变得更加容易。有了自动训练，解决方案从最新数据中学习就不再需要手动训练。如果没有自动训练，则必须手动为解决方案创建新的解决方案版本，这样才能从最新数据中学习。没有自动训练会导致推荐内容过时和转化率下降。有关维护 Amazon Personalize 推荐的更多信息，请参阅维护建议相关性。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface（AWS CLI）或 AWS SDK 来配置自动训练。有关使用控制台配置自动训练的步骤，请参阅创建解决方案（控制台）。

创建解决方案后，记录解决方案 ARN 以备将来使用。通过自动训练，解决方案处于活动状态后，解决方案版本的创建将在一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。训练开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的 Amazon 资源名称（ARN）。要获取其状态，请使用 DescribeSolutionVersion API 操作。

主题

- 指南和要求
- 配置自动训练（AWS CLI）
- 配置自动训练（SDK）

指南和要求

以下是自动训练的准则和要求：

- 只有在自上次训练以来导入了批量交互数据或实时交互数据时，才会进行自动训练。这包括物品交互，或者对于使用 Next-Best-Action 配方的解决方案，包括操作交互数据。
- 每次训练都会考虑包括在训练中的数据集组中的所有数据。有关配置训练中所用列的信息，请参阅配置训练时使用的列。
- 您仍然可以手动创建解决方案版本。
- 解决方案处于活动状态后的一小时内，自动训练会开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。
- 训练计划基于训练开始日期。例如，如果第一个解决方案版本在晚上 7:00 开始训练，而您使用的是每周训练，那么下一个解决方案版本将在一周后的晚上 7:00 开始训练。
- 对于所有配方，我们建议至少每周进行一次训练。您可以指定介于 1 天和 30 天之间的训练频率。默认为每 7 天进行一次训练。
  - 如果您使用 User-Personalization-v2、User-Personalization 或 Next-Best-Action，则解决方案会自动更新为考虑在推荐中加入新物品或操作。自动更新与自动训练不同。自动更新不会创建全新的解决方案版本，模型也不会从最新数据中学习。为了维持您的解决方案，训练频率仍然应该是至少每周一次。有关自动更新的更多信息（包括其他准则和要求），请参阅自动更新。
  - 如果您使用 Trending-Now，则 Amazon Personalize 会在可配置的时间间隔内自动识别交互数据中最热门的物品。Trending-Now 可以通过批量交互数据或流式处理交互数据推荐自上次训练以来添加的物品。您的训练频率仍然应该是至少每周一次。有关更多信息，请参阅 Trending-Now 食谱。
  - 如果您不使用具有自动更新的配方或 Trending-Now 配方，则 Amazon Personalize 仅在下一次训练之后考虑在推荐中加入新物品。例如，如果您使用 Similar-Items 配方并每天添加新物品，则必须使用每日自动训练频率，这些物品才会出现在当天的推荐中。

配置自动训练（AWS CLI）

以下代码向您展示了如何创建每五天自动创建解决方案版本的解决方案。要关闭自动训练，请将 `perform-auto-training` 设置为 `false`。

要更改训练频率，可以修改 `autoTrainingConfig` 中的 `schedulingExpression`。表达式必须采用以下格式：`rate(value unit)`。对于值，指定一个介于 1 和 30 之间的数字。对于单位，请指定 `day` 或 `days`。

有关 `create-solution` 命令的详细说明，请参阅创建解决方案 (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)\"}}"
```

配置自动训练（SDK）

以下代码向您展示了如何使用 AWS SDK 创建带有自动训练的解决方案。该解决方案每五天自动创建一个解决方案版本。要关闭自动训练，请将 `performAutoTraining` 设置为 `false`。

要更改训练频率，可以修改 `autoTrainingConfig` 中的 `schedulingExpression`。表达式必须采用以下格式：`rate(value unit)`。对于值，指定一个介于 1 和 30 之间的数字。对于单位，请指定 `day` 或 `days`。

有关 CreateSolution API 操作的详细说明，请参阅创建解决方案 (AWS SDK)。

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();
```

您可以使用以下 Python 代码来等待自动训练开始。wait_for_training_to_start 方法返回第一个解决方案版本的 ARN。

```
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)
```

配置训练时使用的列

Important

默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

如果您的配方生成了物品建议或用户细分，则可以修改 Amazon Personalize 在创建解决方案版本（训练模型）时考虑的列。

您可以更改训练时使用的列，以控制 Amazon Personalize 在训练模型（创建解决方案版本）时使用的数据。您可以这样做，以试验不同的训练数据组合；也可以排除无有意义数据的列。例如，可能有一列，您只想将其用来筛选建议。您可以将此列排除在训练之外，Amazon Personalize 只有在筛选时才会考虑该列。

您无法排除 EVENT_TYPE 列。默认情况下，Amazon Personalize 使用可在训练时采用的所有列。始终从训练中排除以下数据：

- 具有布尔数据类型的列
- 展示数据
- 既不是分类也不是文本的自定义字符串字段

您不能在训练中包括展示数据，但如果您的应用场景或配方使用了这些数据，Amazon Personalize 会在您获取建议时使用展示数据来指导浏览。

如果您已经创建了一个解决方案，并且想要修改它在训练时使用的列，则可以克隆该解决方案。克隆解决方案时，您可以使用现有解决方案的配置（例如食谱和超参数）作为起点，并根据需要进行任何更改。有关更多信息，请参阅 克隆解决方案（控制台）。

您可以配置 Amazon Personalize 在使用亚马逊个性化控制台、 AWS Command Line Interface (AWS CLI) 或 AWS SDK 进行训练时使用的列。有关使用 Amazon Personalize 控制台选择列的信息，请参阅创建解决方案（控制台）中的高级配置步骤。创建解决方案后，您可以在 Amazon Personalize 控制台的解决方案详情页面或通过 DescribeSolution 操作，查看解决方案使用的列。

主题

- 配置训练时使用的列 (AWS CLI)
- 配置训练时使用的列 (AWS 软件开发工具包)

配置训练时使用的列 (AWS CLI)

要将列排除在训练之外，请在解决方案配置期间，在 `trainingDataConfig` 中提供 `excludedDatasetColumns` 对象。对于每个键，提供数据集类型。对于每个值，提供要排除的列的列表。以下代码显示了如何在通过 AWS CLI创建解决方案时将列排除在训练之外。

```
aws personalize create-solution \
```

```
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--solution-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
 { \"datasetType\" : [ \"column1Name\", \"column2Name\"]}}}"
```

配置训练时使用的列 (AWS 软件开发工具包)

要将列排除在训练之外，请在解决方案配置期间，在 `trainingDataConfig` 中提供 `excludedDatasetColumns` 对象。对于每个键，提供数据集类型。对于每个值，提供要排除的列的列表。以下代码显示了如何在通过 适用于 Python (Boto3) 的 AWS SDK创建解决方案时将列排除在训练之外。

```
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)
```

针对其他目标优化解决方案

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

如果您使用 User-Personalization 配方或 Personalized-Ranking 配方，则除了最大相关性（例如更大限度地提高收入）之外，您还可以针对目标优化 Amazon Personalize 解决方案。

使用物品建议配方时，Amazon Personalize 的主要目标是根据历史和实时物品交互数据，为您的用户预测最相关的物品。这些是您的用户最有可能与之交互的物品（例如，他们最有可能单击的物品）。如果您还有其他目标，例如最大限度地提高流时长或增加收入，则可以创建一个解决方案，根据相关性和目标生成建议。

要针对其他目标优化解决方案，请使用 User-Personalization 食谱或 Personalized-Ranking 食谱创建一个新的解决方案，然后在物品数据集中选择与您的目标相关的数值元数据列。在生成建议时，Amazon Personalize 会更加重视此列数据中具有更高值的物品。例如，您可以选择 VIDEO_LENGTH 列来最大限度地提高流时长，或者选择 PRICE 列来最大限度地提高收入。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs。有关使用 Amazon Personalize 控制台的信息，请参阅创建解决方案（控制台）。

主题

- 指南和要求
- 平衡目标重点和相关性
- 衡量优化性能
- 优化解决方案 (AWS CLI)
- 优化解决方案 (AWS SDKs)
- 示例 Jupyter 笔记本

指南和要求

目标要求如下：

- 您只能为目标选择一列。
- 列在架构中必须具有数字类型。
- 列在架构中不能有 `null` 类型。

有关架构和数据类型的更多信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。

平衡目标重点和相关性

当更多地根据目标（而不是相关性）来推荐物品时，可能会进行权衡取舍。例如，如果您想通过建议来增加收入，则仅针对昂贵物品的建议可能会降低物品与用户的相关性，并降低用户参与度和转化率。

要配置相关性和目标之间的平衡，请在创建解决方案时选择以下一个目标敏感度级别：

- 关闭：Amazon Personalize 主要使用物品交互数据为用户预测最相关的物品。
- 低：Amazon Personalize 不太重视您的目标。通过物品交互数据实现的相关性更为重要。
- 中等：Amazon Personalize 通过物品交互数据对您的目标和相关性给予同等重视。
- 高：Amazon Personalize 更加重视您的目标。通过物品交互数据实现的相关性不太重要。

衡量优化性能

当您为具有优化目标的解决方案创建解决方案版本（训练模型）时，Amazon Personalize 会生成 `average_rewards_at_k` 指标。`average_rewards_at_k` 的分数可指明，解决方案版本在实现目标方面的性能如何。为了计算该指标，Amazon Personalize 按如下方式计算每位用户的奖励：

```
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
```

最终 `average_rewards_at_k` 是所有 `rewards_per_user` 的平均值，即标准化为小于或等于 1 且大于 0 的十进制值。该值越接近 1，表示您期望从建议中获得的每个用户平均收益就越多。

例如，如果您的目标是最大限度地提高单击收入，则 Amazon Personalize 将按以下方法计算每个用户分数：将用户单击的前 25 个最昂贵建议物品产生的总收入除以用户单击的所有推荐物品产生的总收入。然后，Amazon Personalize 会返回所有用户分数的标准化平均值。`average_rewards_at_k` 越接近 1，表示您期望从建议中获得的每个用户平均收入就越多。

有关生成指标的更多信息，请参阅通过指标评估 Amazon Personalize 解决方案版本。

优化解决方案 (AWS CLI)

您只能使用 User-Personalization 或 Personalized-Ranking 食谱针对目标进行优化。要使用针对其他目标优化解决方案 AWS CLI，请创建一个新的解决方案并使用`solutionConfig`对象中的`optimizationObjective`密钥指定您的目标细节。`optimizationObjective` 包含以下字段：

- `itemAttribute`：指定物品数据集中与您的目标相关的数值元数据列的名称。

- objectiveSensitivity：指定解决方案在生成建议时对您的目标的重视程度。目标敏感度级别配置了 Amazon Personalize 如何通过物品交互数据根据目标和相关性来平衡建议的物品。objectiveSensitivity 可以是 OFF、LOW、MEDIUM 或 HIGH。有关更多信息，请参阅 平衡目标重点和相关性。

以下是 create-solution AWS CLI 命令的示例。将 solution name、dataset group arn 和 recipe arn 值替换为您自己的值。

对于 optimizationObjective，将 COLUMN_NAME 替换为物品数据集中与您的目标相关的数值元数据列名称。对于 objectiveSensitivity，请指定关闭、低、中或高。

```
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\"}}"
```

解决方案准备就绪后，创建一个新的解决方案版本（有关示例命令，请参阅创建解决方案 (AWS CLI)）。创建解决方案版本后，可以通过解决方案版本指标查看优化性能。请参阅衡量优化性能。

优化解决方案 (AWS SDKs)

您只能使用 User-Personalization 或 Personalized-Ranking 食谱针对目标进行优化。

要使用针对其他目标优化解决方案 AWS SDKs，请创建一个新的解决方案并使用解决方案solutionConfig对象中的optimizationObjective密钥指定您的目标细节。optimizationObjective 包含以下字段：

- itemAttribute：指定数据集组物品数据集中与您的目标相关的数值元数据列的名称。
- objectiveSensitivity：指定解决方案在生成建议时对您的目标的重视程度。目标敏感度级别配置了 Amazon Personalize 如何通过物品交互数据根据目标和相关性来平衡建议的物品。objectiveSensitivity 可以是 OFF、LOW、MEDIUM 或 HIGH。有关更多信息，请参阅 平衡目标重点和相关性。

使用以下代码通过 适用于 Python (Boto3) 的 AWS SDK 或 AWS SDK for Java 2.x创建具有其他目标的解决方案。

解决方案准备就绪后，创建一个新的解决方案版本（有关示例代码，请参阅创建解决方案版本 (AWS SDK)）。创建解决方案版本后，可以通过解决方案版本指标查看优化性能。请参阅衡量优化性能。

SDK for Python (Boto3)

要创建针对其他目标进行了优化的解决方案，请使用以下 create_solution 方法。将 solution name、dataset group arn 和 recipe arn 值替换为您自己的值。

对于 optimizationObjective，将 COLUMN_NAME 替换为物品数据集中与您的目标相关的数值元数据列名称。对于 objectiveSensitivity，请指定关闭、低、中或高。

```
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

要创建针对其他目标进行了优化的解决方案，请使用以下 createPersonalizeSolution 方法并将以下内容作为参数传递：Amazon Personalize 服务客户端、数据集组的 Amazon 资源名称 (ARN)、解决方案名称、食谱 ARN、物品属性和目标敏感度级别。

```
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();
```

示例 Jupyter 笔记本

有关演示如何创建针对其他基于目标的项目元数据进行优化的解决方案的 Jupyter 笔记本示例，请参阅 Amazon Personalize 示例存储库的 objective_ optimation 文件夹 GitHub

使用事件配置优化解决方案

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

如果您使用了 User-Personalization-v2 recipe 或 Personalized-Ranking-v2 配方，则可以使用事件配置优化 Amazon Personalize 解决方案。

使用项目建议配方时，Amazon Personalize 的主要目标是根据历史和实时项目交互数据，为您的用户预测最相关的项目。但是，交互可能包含其他信息，例如用户是点击还是购买了某个项目。您可以通过

记录事件的类型（事件类型和事件值数据）来记录此内容。配置解决方案时，您可以让它为不同的交互事件类型赋予不同的权重。例如，您可以配置一个解决方案，为 purchase 事件赋予高于 click 事件的权重。

要让解决方案为不同的事件类型赋予不同的权重，您需要在解决方案的事件配置中指定事件类型及其对应的权重。此外，您还可以设置事件值阈值以排除事件值低于该阈值的交互。例如，如果 EVENT_TYPE 为观看 的事件的 EVENT_VALUE 数据是用户观看的视频的百分比，如果您将事件值阈值设置为 0.5，将事件类型设置为观看，则 Amazon Personalize 将仅使用 EVENT_VALUE 大于或等于 0.5 的观看 交互事件来训练模型。

与事件类型相关的权重将决定事件类型的重要性。权重较高的事件类型将使经过训练的模型更有可能推荐与该事件类型交互的项目。例如，如果您为“购买”赋予了高于“点击”的权重，并且模型得知某用户将点击项目 B 或利用自己的交互历史购买项目 C，那么模型将提高项目 C 的排名。

要使用事件配置优化解决方案，您需要使用 Personalization-v2 配方或 Personalized-Ranking-v2 配方创建一个新解决方案，并指定事件配置。您还可以使用事件配置更新现有解决方案（更新解决方案以更改其自动训练配置）。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK。有关使用 Amazon Personalize 控制台的信息，请参阅创建解决方案（控制台）。

主题

- 指南和要求
- 使用事件权重配置来衡量性能
- 优化解决方案 (AWS CLI)
- 优化解决方案 (AWS SDK)

指南和要求

下面是事件配置的准则和要求：

- 要为不同的事件类型配置权重，您的项目交互数据集必须有一个 EVENT_TYPE 列和一个 EVENT_VALUE 列（可选）。
- 您可以在配置中指定一个事件参数列表。包含您希望在创建解决方案时考虑的所有事件类型。您可以指定最多 10 种不同的事件类型。
- 您可以为每种事件类型指定事件权重。事件权重必须介于 0.0 和 1.0 之间。只有事件类型之间的权重比率才重要。例如，将事件类型“购买”的权重设置为 0.3 并将事件类型“点击”的权重设置为 0.1，其效果与将“购买”的权重设置为 0.6 并将“点击”的权重设置为 0.2 相同。

- 您可以使用 UpdateSolution API 操作更新现有解决方案的事件配置。

使用事件权重配置来衡量性能

当您为具有事件配置的解决方案创建解决方案版本（训练模型）时，Amazon Personalize 会生成 normalized_discounted_cumulative_gain_with_event_weights_at_k 指标。normalized_discounted_cumulative_gain_with_event_weights_at_k 的分数表示在考虑为每个事件类型设置的事件权重的情况下，解决方案版本的表现情况。

它与 k 阶标准化折损累积增益（NDCG）类似，只不过对每个正确预测的奖励分数会进行加权处理。与之不同的是，在原始 K 阶 NDCG 中，每个正确预测的权重均为 1。例如，如果“购买”权重为 0.3，“点击”权重为 0.1，正确预测“购买”项目将获得 1.5 的奖励分数，而预测“点击”项目将获得 0.5 的奖励分数。

有关生成指标的更多信息，请参阅通过指标评估 Amazon Personalize 解决方案版本。

优化解决方案 (AWS CLI)

您可以将 User-Personalization-v2 或 Personalized-Ranking-v2 配方与事件配置结合使用来进行优化。

要借助 AWS CLI 使用事件配置来优化解决方案，请创建一个新解决方案，并在 solutionConfig 对象中使用 eventsConfig 键指定事件配置详细信息。eventsConfig 有一个键 eventParametersList，您可在该键下指定最多 10 个 eventParameters。每个 eventParameter 都包含以下字段：

- eventType：指定您在创建解决方案时要考虑的事件类型。
- eventValueThreshold：指定事件值阈值。只有事件值大于或等于此阈值的事件才会被考虑用于创建解决方案。
- weight：为每种事件类型指定权重。权重越高意味着事件类型对创建的解决方案越重要。

以下是 create-solution AWS CLI 命令的示例。将 solution name、dataset group arn 和 recipe arn 值替换为您自己的值。

```
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}]}"
```

解决方案准备就绪后，创建一个新的解决方案版本（有关示例命令，请参阅创建解决方案 (AWS CLI)）。创建解决方案版本后，可以通过解决方案版本指标查看优化性能。请参阅衡量优化性能。

优化解决方案 (AWS SDK)

您可以将 User-Personalization-v2 或 Personalized-Ranking-v2 配方与事件配置结合使用来进行优化。

要借助 AWS SDK 使用事件配置来优化解决方案，请创建一个新解决方案，并在 `solutionConfig` 对象中使用 `eventsConfig` 键指定事件配置详细信息。`eventsConfig` 有一个键 `eventParametersList`，您可在该键下指定最多 10 个 `eventParameters`。每个 `eventParameter` 都包含以下字段：

- eventType：指定您在创建解决方案时要考虑的事件类型。
- eventValueThreshold：指定事件值阈值。只有事件值大于或等于此阈值的事件才会被考虑用于创建解决方案。
- weight：为每种事件类型指定权重。权重越高意味着事件类型对创建的解决方案越重要。

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 "";
```

解决方案准备就绪后，创建一个新的解决方案版本（有关示例命令，请参阅创建解决方案 (AWS SDK)）。创建解决方案版本后，可以通过解决方案版本指标查看优化性能。请参阅衡量优化性能。

超级参数和 HPO

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

您可以在训练之前指定超参数，以便针对您的特定使用案例优化经过训练的模型。这与模型参数相反，后者的值在训练过程中确定。

超参数是使用 `algorithmHyperParameters` 键指定的，该键是传递到 CreateSolution 操作的 SolutionConfig 对象的一部分。

`CreateSolution` 请求的精简版本如下。该示例包括 `solutionConfig` 对象。您使用 `solutionConfig` 来覆盖食谱的默认参数。

```
{
  "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"
          }
```

```
      },
  },
}
```

使用不同超级参数的不同食谱。有关可用的超级参数，请参阅选择食谱中的各个配方。

启用超参数优化

超参数优化 (HPO) 或调整是针对特定学习目标选择最佳超参数的任务。最佳超级参数是通过运行使用来自指定可能性范围的不同值的很多训练任务确定的。

使用User-Personalization-v2和 Personalized-Ranking-v2，如果您开启自动训练，Amazon Personalize 会每 90 天自动执行 HPO。如果没有开启自动训练，就不会执行 HPO。对于所有其他配方，必须启用 HPO。要使用 HPO，请将 performHPO 设置为 true，并包含 hpoConfig 对象。

超参数可以是分类值、连续值或整数值。hpoConfig 对象具有与所有这些类型对应的键，您可以在其中指定超级参数及其范围。您必须在请求中提供每种类型，但是如果食谱没有某个类型的参数，则您可以将其留空。例如， User-Personalization 没有连续类型的可调超参数。因此，对于 continousHyperParameterRange，您可以传递一个空数组。

以下代码显示如何使用创建启用 HPO 的 适用于 Python (Boto3) 的 AWS SDK解决方案。本示例中的解决方案使用 User-Personalization 食谱食谱，并将 HPO 设置为 true。代码为 hidden_dimension、categoricalHyperParameterRanges 和 integerHyperParameterRanges 提供值。continousHyperParameterRange 为空，hpoResourceConfig 设置为 maxNumberOfTrainingJobs 和 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"
          }
        }
    }
)
```

有关 HPO 的更多信息，请参阅自动模型优化。

查看超参数

您可以通过调用 DescribeSolution 操作来查看解决方案的超参数。下面的示例显示了 `DescribeSolution` 输出。创建解决方案版本（训练模型）后，您还可以通过 DescribeSolutionVersion 操作查看超参数。

```
{
  "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"
  }
}
```

选择用于训练的物品交互数据

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

您可以在物品交互数据集中选择 Amazon Personalize 在创建解决方案版本（训练模型）时使用的事件。通过在训练之前选择物品交互数据，您可以只使用数据的相关子集进行训练，或者去除噪声来训练更优化的模型。有关物品交互数据集的更多信息，请参阅物品交互数据。

> **Note**
>
> 如果您使用 User-Personalization-v2 或 Personalized-Ranking-v2，则您的训练费用将基于您的项目交互数据，然后再按事件类型或值进行筛选。有关定价的更多信息，请参阅 Amazon Personalize 定价。

您可以按如下方式选择物品交互数据：

- 根据类型选择记录 - 配置解决方案时，如果您的物品交互数据集在 EVENT_TYPE 列中包含事件类型，则您可以选择指定要在训练中使用的事件类型。例如，如果您的项目交互数据集包含购买、点击和观看事件类型，并且您希望 Amazon Personalize 仅使用观看事件来训练模型，则在配置解决方案时，您需要提供观看作为 Amazon Personalize 在训练中使用的 `event type`。

  如果您有多个事件类型并使用 User-Personalization-v2 配方或配 Personalized-Ranking-v2 方，则在配置自定义解决方案时，可以为不同的类型指定不同的权重。例如，您可以配置一个解决方案，为购买事件赋予高于点击事件的权重。有关更多信息，请参阅 使用事件配置优化解决方案。

  如果您的物品交互数据集在 EVENT_TYPE 列中有多个事件类型，并且您在配置解决方案时未提供某个事件类型，则 Amazon Personalize 会使用所有物品交互数据进行训练，无论其类型如何，权重均相同。
- 根据类型和值选择记录 - 配置解决方案时，如果物品交互数据集包含 EVENT_TYPE 和 EVENT_VALUE 字段，则您可以将特定值设置为阈值，以从训练中排除记录。例如，如果 EVENT_TYPE 为观看 的事件的 EVENT_VALUE 数据是用户观看的视频的百分比，如果您将事件值

阈值设置为 0.5，将事件类型设置为观看，则 Amazon Personalize 将仅使用 EVENT_VALUE 大于或等于 0.5 的观看 交互事件来训练模型。

以下代码演示如何使用创建解决方案，该解决方案仅使用用户观看了一半以上视频watch的事件。 适用于 Python (Boto3) 的 AWS SDK

```
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'])
```

克隆解决方案（控制台）

创建新解决方案时，您可以使用 Amazon Personalize 控制台来克隆解决方案。克隆解决方案时，您可以使用现有解决方案的配置（例如食谱和超参数）作为起点，并根据需要进行任何更改。如果您想对解决方案进行一项更改，但保留所有其他属性不变，则这样做很有用。例如，向数据集添加一列新的训练数据。在这种情况下，您可以克隆解决方案，为解决方案命名，更改训练时使用的列，并保持所有其他属性不变。

克隆解决方案

要克隆解决方案，您需选择现有解决方案，然后选择克隆解决方案选项。随后，为新解决方案命名，并修改相关字段。

克隆解决方案

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 选择自定义资源，然后选择解决方案。
4. 选择要克隆的解决方案。
5. 选择操作，然后选择克隆解决方案。
6. 为新解决方案命名。
7. 对解决方案详细信息和高级配置进行任何更改。Amazon Personalize 会使用现有解决方案中的值预填充这些字段。有关每个字段的信息，请参阅在 Amazon Personalize 中配置自定义解决方案。

更新解决方案以更改其自动训练配置

创建解决方案后，您可以更改其自动训练配置和事件配置：

- 您可以打开或关闭自动训练，也可以更改训练频率。
  - 如果开启自动训练，则第一次自动训练将在解决方案更新完成后的一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。
  - 如果您修改解决方案的训练频率，则训练计划会重置，并且新的解决方案版本将在一小时内开始训练。解决方案版本的创建继续按新的频率进行，即第 1 天是更新解决方案的那一天。
- 您可以更新解决方案事件配置。如果解决方案已指定一个事件配置，则新的事件配置将替换原来的配置。有关针对事件配置优化解决方案的信息，请参阅使用事件配置优化解决方案。

您可以使用 Amazon Personalize 控制台或 AWS 软件开发工具包更新解决方案。 AWS Command Line Interface解决方案更新要几分钟。在更新过程中，您可以为解决方案创建解决方案版本，但不能删除解决方案。在更新完成之前，解决方案使用之前的配置。有关自动训练的更多信息，请参阅配置自动训练。

主题

- 更新解决方案（控制台）
- 更新解决方案 (AWS CLI)
- 更新解决方案 (AWS 软件开发工具包)

更新解决方案（控制台）

要在控制台中更新解决方案，请导航到解决方案，选择更新，然后指定要使用的新配置。

配置解决方案

1. 打开位于的 Amazon Personalize 控制台 https://console.aws.amazon.com/personalize/home，然后登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 在导航窗格中，选择自定义资源，然后选择解决方案和配方。
4. 选择您的解决方案，然后选择右上角的更新。
5. 在自动训练中，修改解决方案是否使用自动训练。如果自动训练已开启，则可以更改 `Automatic training frequency`。默认训练频率为每 7 天一次。
6. 选择更新解决方案。您可以在解决方案的详细信息页面上找到解决方案更新的状态。

更新解决方案 (AWS CLI)

要使用更新解决方案 AWS Command Line Interface，请使用`update-solution`命令。此命令使用 UpdateSolution API 操作。以下代码展示了如何更新解决方案以使用训练频率为 5 天的自动训练。要关闭自动训练，请指定 `--no-perform-auto-training` 并省略 `solution-update-config`。

默认训练频率为每 7 天一次。表达式必须采用以下格式：`rate(value unit)`。对于值，指定一个介于 1 和 30 之间的数字。对于单位，请指定 `day` 或 `days`。

```
aws personalize update-solution \
--solution-arn solution ARN \
--perform-auto-training \
--solution-update-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5
 days)\"}}"
```

要获取更新的状态，请使用 `describe-solution` 命令（使用 DescribeSolution API 操作），并在 `latestSolutionUpdate` 中找到更新状态。

更新解决方案 (AWS 软件开发工具包)

要使用 AWS 软件开发工具包更新解决方案，请使用 UpdateSolution API 操作。以下代码向您展示了如何使用更新解决方案 适用于 Python (Boto3) 的 AWS SDK 以使用训练频率为 5 天的自动训练。该代码通过 DescribeSolution API 操作获取更新的状态。

默认训练频率为每 7 天一次。表达式必须采用以下格式：rate(*value unit*)。对于值，指定一个介于 1 和 30 之间的数字。对于单位，请指定 day 或 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}")
```

手动创建解决方案版本

完成在 Amazon Personalize 中配置自定义解决方案后，就可以开始训练了：

- 如果您的解决方案使用自动训练，则解决方案会按照您指定的训练频率为您创建解决方案版本。默认情况下，所有新解决方案都使用自动训练，每 7 天创建一个新的解决方案版本。您仍然可以手动创建解决方案版本。有关更多信息，请参阅 配置自动训练。
- 如果您关闭解决方案的自动训练，或者想要手动训练，则可以手动创建解决方案版本。解决方案版本 是指经过训练的机器学习模型。您可以使用控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 创建解决方案版本。如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以使用 the section called “StopSolutionVersionCreation” 操作来停止解决方案版本的创建过程。请参阅停止创建解决方案版本。

如果由于错误而无法完成训练，则您无需支付训练费用。如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以停止解决方案版本的创建过程。要停止

创建解决方案版本，请导航至解决方案版本详细信息页面，然后选择停止。有关更多信息，请参阅 停止创建解决方案版本。

主题

- 创建解决方案版本（控制台）
- 创建解决方案版本 (AWS CLI)
- 创建解决方案版本 (AWS SDK)

创建解决方案版本（控制台）

要使用 Amazon Personalize 控制台手动创建新的解决方案版本，请从解决方案的详细信息页面开始训练。

创建新解决方案版本

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 导航到数据集组页面，然后选择包含新解决方案的数据集组。
3. 在导航窗格中的自定义资源下，选择解决方案和配方。
4. 在解决方案和食谱页面上，选择要创建解决方案版本的解决方案。
5. 在解决方案概述页面上，选择创建解决方案版本以开始训练新模型。

在解决方案详细信息页面上，您可以在解决方案版本部分跟踪训练进度。训练完成后，状态会变为活动，您可以使用 Amazon Personalize 提供的指标对其进行评估。有关更多信息，请参阅 通过指标评估 Amazon Personalize 解决方案版本。

当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

- 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
- 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

创建解决方案版本 (AWS CLI)

当您的解决方案处于活动状态时，通过运行以下命令来训练模型。将 `solution arn` 替换为 在 Amazon Personalize 中配置自定义解决方案 中的解决方案 Amazon 资源名称 (ARN)。

```
aws personalize create-solution-version \
  --solution-arn solution arn
```

这将显示解决方案版本 ARN，例如：

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>"
}
```

通过使用 `describe-solution-version` 命令来检查解决方案版本的训练状态。提供上一步中返回的解决方案版本 ARN。有关 API 的更多信息，请参阅DescribeSolutionVersion。

```
aws personalize describe-solution-version \
  --solution-version-arn solution version arn
```

将显示解决方案版本的属性和训练 `status`。最初，状态显示为 CREATE PENDING，例如：

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

训练完成后，`status` 会变为 `ACTIVE`，您可以使用 Amazon Personalize 提供的指标对其进行评估。有关更多信息，请参阅 通过指标评估 Amazon Personalize 解决方案版本。如果由于错误而无法完成训练，则您无需支付训练费用。

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以使用 StopSolutionVersionCreation 操作来停止解决方案版本的创建过程。请参阅停止创建解决方案版本。

当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

- 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
- 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

创建解决方案版本 (AWS SDK)

当您的解决方案处于活动状态时，使用以下代码创建解决方案版本。指定 在 Amazon Personalize 中配置自定义解决方案 中的 Amazon 资源名称（ARN）。使用 DescribeSolutionVersion 操作来检索解决方案版本的状态。

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();
```

要检查当前解决方案版本状态，请调用 DescribeSolutionVersion 操作并传递从 CreateSolutionVersion 操作返回的解决方案版本的 ARN。训练完成后，status 会变为 ACTIVE，您可以使用 Amazon Personalize 提供的指标对其进行评估。有关更多信息，请参阅 通过指标评估 Amazon Personalize 解决方案版本。如果由于错误而无法完成训练，则您无需支付训练费用。

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以使用 StopSolutionVersionCreation 操作来停止解决方案版本的创建过程。请参阅停止创建解决方案版本。

当解决方案版本处于活动状态时，就可以使用它来获得推荐。如何使用活跃解决方案版本取决于获取建议的方式：

- 要获得实时建议，必须通过 Amazon Personalize 市场活动部署活跃解决方案版本。您可以使用该市场活动为用户获取建议。请参阅通过市场活动部署 Amazon Personalize 解决方案版本。
- 对于批量建议，在创建批量推理作业或批量细分作业时指定一个活跃解决方案版本。请参阅获取批量物品推荐或获取批量用户细分。

停止创建解决方案版本

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以使用 Amazon Personalize 控制台或 StopSolutionVersionCreation 操作来停止创建解决方案版本（停止训练模型）。解决方案版本停止后，就无法继续创建。您需要为解决方案版本停止创建之前使用的资源付费。

停止创建解决方案版本会结束模型训练，但不会删除解决方案版本。您仍然可以在 Amazon Personalize 控制台和 DescribeSolutionVersion 操作中查看解决方案版本详情。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或，停止解决方案版本的创建过程 AWS SDKs。

主题

- 停止创建解决方案版本（控制台）
- 停止创建解决方案版本 (AWS CLI)
- 停止创建解决方案版本 (AWS SDKs)

停止创建解决方案版本（控制台）

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以停止创建解决方案版本（停止训练模型）。

停止创建解决方案版本（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择包含要停止的解决方案版本的数据集组。
3. 在导航窗格中，选择解决方案和食谱。
4. 在解决方案和食谱页面上，选择包含要停止的解决方案版本的解决方案。
5. 在解决方案版本中，选择要停止的解决方案版本。
6. 在解决方案版本详细信息页面上，选择停止创建。根据解决方案版本的原始状态，解决方案版本状态会发生如下变化：
   - CREATE_PENDING 更改为 CREATE_STOPPED。
   - CREATE_IN_PROGRESS 更改为 CREATE_STOPPING，然后更改为 CREATE_STOPPED。

停止创建解决方案版本 (AWS CLI)

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以停止创建解决方案版本（停止训练模型）。通过以下 `stop-solution-version-creation` 命令停止使用 AWS CLI创建解决方案版本。将 `solution version arn` 替换为要停止的解决方案版本的 Amazon 资源名称 (ARN)。您需要为解决方案版本停止创建之前使用的资源付费。

```
aws personalize stop-solution-version-creation \
    --solution-version-arn solution version arn
```

通过使用 `describe-solution-version` 命令来检查解决方案版本的训练状态。

```
aws personalize describe-solution-version \
    --solution-version-arn solution version arn
```

根据解决方案版本的原始状态，解决方案版本状态会发生如下变化：

- CREATE_PENDING 更改为 CREATE_STOPPED。
- CREATE_IN_PROGRESS 更改为 CREATE_STOPPING，然后更改为 CREATE_STOPPED

停止创建解决方案版本 (AWS SDKs)

如果解决方案版本的状态为 CREATE_PENDING 或 CREATE_IN_PROGRESS，则您可以停止创建解决方案版本（停止训练模型）。以下代码显示如何停止使用 适用于 Python (Boto3) 的 AWS SDK 或创建解决方案版本 AWS SDK for Java 2.x。您需要为解决方案版本停止创建时使用的资源付费。

SDK for Python (Boto3)

使用以下 `stop_solution_version_creation` 方法停止创建解决方案版本。将 `solution_version_arn` 替换为要停止的解决方案版本的 Amazon 资源名称 (ARN)。该方法使用 DescribeSolutionVersion 操作来检索解决方案版本的状态。

```
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

使用以下 `stopSolutionVersionCreation` 方法停止创建解决方案版本。将 Amazon Personalize 服务客户端和要停止创建的解决方案版本的 Amazon 资源名称 (ARN) 作为参数传递。以下代码使用 DescribeSolutionVersion 操作来检索解决方案版本的状态。

```
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);
}
```

根据解决方案版本的原始状态，解决方案版本状态会发生如下变化：

- CREATE_PENDING 更改为 CREATE_STOPPED。

- CREATE_IN_PROGRESS 更改为 CREATE_STOPPING，然后更改为 CREATE_STOPPED。

通过指标评估 Amazon Personalize 解决方案版本

您可以通过离线和在线指标来评估解决方案版本的性能。在线指标 是您在用户与实时建议的交互中观察到的实证结果。例如，您可以记录用户在浏览您的目录时的点击率。您负责生成和记录任何在线指标。

离线指标 是 Amazon Personalize 在您训练解决方案版本时生成的指标。您可使用离线指标来评估模型的性能。您可以查看修改解决方案超参数产生的影响，并且可以比较同一数据集组中使用相同数据但针对不同配方进行训练的模型的结果。

避免比较使用不同数据训练的不同解决方案版本的指标。指标的差异可能源于数据的差异，而不是源于模型性能。例如，您可能有一个数据集组，其中包含每个用户的稀疏 `purchase` 事件数据，而另一个数据集组则包含可靠的 `view` 事件数据。根据像 `precision at K` 这样的指标，由于交互次数较多，根据查看事件数据训练的解决方案版本可能会错误地表现得更好。

为了获得性能指标，Amazon Personalize 将输入的交互数据拆分为训练集、测试集以及验证集（验证集适用于 PERSONALIZED_ACTIONS）。拆分取决于您选择的食谱类型：

- 对于 USER_SEGMENTATION 食谱，训练集由每个用户 80% 的交互数据组成，测试集由每个用户 20% 的交互数据组成。
- 对于所有其他食谱类型，训练集由 90% 的用户及其交互数据组成。测试集由剩余的 10% 的用户及其交互数据组成。

然后，Amazon Personalize 使用训练集创建解决方案版本。训练完成后，Amazon Personalize 会向新解决方案版本提供测试集中每位用户 90% 的最早数据作为输入。然后，Amazon Personalize 通过将解决方案版本生成的建议与测试集中每位用户 10% 的最新数据中的实际交互进行比较，计算指标。

为了生成基准以进行比较，我们建议使用热门程度-计数食谱，该食谱会推荐前 K 项最受欢迎的物品。

主题

- 检索解决方案版本指标
- 指标定义
- 示例
- 其他资源

检索解决方案版本指标

创建解决方案版本后，您可以使用指标来评估其性能。您可以使用 Amazon Personalize 控制台AWS Command Line Interface（AWS CLI）和，检索解决方案版本的指标AWSSDKs。

主题

- 检索解决方案版本指标（控制台）
- 检索解决方案版本指标 (AWS CLI)
- 检索解决方案版本指标 (AWSSDKs)

检索解决方案版本指标（控制台）

要在控制台中查看推荐器指标，请导航到解决方案版本的详细信息页面。

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择您的自定义数据集组。
3. 在导航窗格中，选择自定义资源，然后选择解决方案和食谱。
4. 选择您的解决方案。
5. 在解决方案版本中，选择您的解决方案版本以查看其详细信息页面。指标列在底部窗格的解决方案版本指标选项卡上。有关指标的定义，请参阅指标定义。

   既然您评估了解决方案版本，就可以通过最适合您使用案例的指标来部署解决方案版本，以创建市场活动。有关部署解决方案的更多信息，请参阅通过市场活动部署 Amazon Personalize 解决方案版本。

检索解决方案版本指标 (AWS CLI)

您通过调用 GetSolutionMetrics 操作为特定解决方案版本检索指标。以下代码显示了如何使用 AWS CLI 检索指标。

```
personalize get-solution-metrics --solution-version-arn solution version ARN
```

以下是使用带有其他优化目标的 User-Personalization 食谱创建的解决方案版本的输出示例。

```
{
```

```
    "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
    }
}
```

有关每个指标的说明，请参阅指标定义。既然您评估了解决方案版本，就可以通过最适合您使用案例的指标来部署解决方案版本，以创建市场活动。有关部署解决方案的更多信息，请参阅通过市场活动部署 Amazon Personalize 解决方案版本。

检索解决方案版本指标 (AWSSDKs)

您通过调用 GetSolutionMetrics 操作为特定解决方案版本检索指标。使用以下代码来检索指标。

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);
    }
}
```

以下是使用带有其他优化目标的 User-Personalization 食谱创建的解决方案版本的输出示例。

```
{
    "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
    }
}
```

有关每个指标的说明，请参阅指标定义。既然您评估了解决方案版本，就可以通过最适合您使用案例的指标来部署解决方案版本，以创建市场活动。有关部署解决方案的更多信息，请参阅通过市场活动部署 Amazon Personalize 解决方案版本。

指标定义

下文使用以下术语描述了 Amazon Personalize 为解决方案版本生成的指标：

- 相关建议 是针对用户实际与之交互的物品的建议。这些物品来自测试集中每个用户 10% 的最新交互数据。
- 排名 是指推荐项在推荐列表中的位置。位置 1（列表的顶部）被假定为与用户最相关。

对于每个指标，数字越大（越接近 1）越好。要深入了解相关信息，请参阅其他资源中列出的资源。

覆盖

覆盖率的值告诉您，Amazon Personalize 可能推荐的唯一物品（针对物品推荐）、操作（针对操作推荐）或用户（针对用户细分推荐）在您的数据集中的唯一记录总数中所占的比例。

覆盖率分数越高，意味着 Amazon Personalize 建议的目录越多，而不是重复记录相同的物品。对于具有物品浏览功能的配方（例如 User-Personalization），其覆盖率比不具备该功能的配方（例如 Similar-Items）的覆盖率更高。

前 25 项中平均倒数排名

该指标可指明，模型在排名靠前的位置生成相关物品建议的能力。

如果您要为用户生成物品搜索结果，且不希望用户选择列表中排名靠后的物品，则可以选择前 25 项中平均倒数排名较高的模型。例如，用户经常在搜索结果中选择第一个烹饪食谱。Amazon Personalize 不会为 PERSONALIZED_ACTIONS 或 USER_SEGMENTATION 配方生成此指标。

Amazon Personalize 使用建议请求的平均倒数排名分数来计算该指标。每个倒数排名分数的计算方法如下：`1 / the rank of the highest item interacted with by the user`，其中可能的总排名为 25。用户与之交互的其他排名较低的物品将被忽略。如果用户选择了第一个物品，则分数为 1。如果没有选择任何物品，则分数为 0。

例如，您可以向三个不同的用户分别显示 25 条建议：

- 如果用户 1 单击排名为 4 的物品和排名为 10 的物品，则其倒数排名分数为 1/4。
- 如果用户 2 单击排名为 2 的物品、排名为 4 的物品以及排名为 12 的物品，则其倒数排名分数为 1/2。
- 如果用户 3 单击排名为 6 的单个物品，则其倒数排名分数为 1/6。

所有建议请求的平均倒数排名（在本例中为 3）的计算公式为 `(1/4 + 1/2 + 1/6) / 3 = .3056`。

K (5/10/25) 项标准化折扣累计增益 (NDCG)

该指标可指明模型对物品或操作建议的排名，其中 K 是样本大小，即 5、10 或 25 条建议。如果您最感兴趣的是推荐的排名，而不仅仅是排名最高的物品或操作，则此指标非常有用（有关此信息，请参阅 `mean reciprocal rank at 25`）。例如，如果您的应用程序在轮播中一次最多显示 10 部电影，则 `NDCG at 10` 分数将非常有用。

Amazon Personalize 根据测试集中每位用户的排名位置为建议分配权重，来计算 NDCG。每条建议被折扣一倍（给定较低的权重），具体取决于其位置。最终指标是测试集中所有用户的平均 `NDCG at K`。`NDCG at K` 假定列表中排名较低的推荐与列表中排名较高的推荐相比相关性更低。

Amazon Personalize 使用加权系数 1/log(1 + position)，其中列表的顶部是位置 1。

K 项精度

该指标可指明，根据样本大小 K（5、10 或 25）条建议，模型建议的相关性。

Amazon Personalize 按照以下方法计算此指标：测试集中每个用户前 K 条建议中的相关建议数量除以 K，其中 K 为 5、10 或 25。最终指标是测试集中所有用户的平均值。

例如，如果您向用户推荐 10 个物品，而用户与其中 3 个物品交互，则 K 项精度为：3 个正确预测的物品除以总计 10 个建议物品 (3 / 10 = .30)。

此指标奖励相关物品的精准建议。分数越接近 1，模型就越精确。

精度

如果您使用 Next-Best-Action配方训练解决方案版本，Amazon Personalize 会生成一个precision指标，而不是precision at K。该指标告诉您，模型在预测用户实际采取的操作方面的表现如何。

为了计算 precision，对于数据集中的每个操作，Amazon Personalize 将经正确预测会采取该操作的用户数除以建议该操作的总次数。然后，Amazon Personalize 计算数据集中所有操作的平均值。

例如，如果向 100 个用户建议了一个操作，60 个用户采取了该操作，而 40 个用户没有采取该操作，则该操作的 precision 为 60 / 100 = .60。然后，Amazon Personalize 将此计算应用于所有操作并返回平均值。

此指标奖励相关操作的精准建议。分数越接近 1，模型就越精确。

average_rewards_at_k

当您为具有优化目标的解决方案创建解决方案版本（训练模型）时，Amazon Personalize 会生成 average_rewards_at_k 指标。average_rewards_at_k 的分数可指明，解决方案版本在实现目标方面的性能如何。为了计算该指标，Amazon Personalize 按如下方式计算每位用户的奖励：

```
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
```

最终 average_rewards_at_k 是所有 rewards_per_user 的平均值，即标准化为小于或等于 1 且大于 0 的十进制值。该值越接近 1，表示您期望从建议中获得的每个用户平均收益就越多。

例如，如果您的目标是最大限度地提高单击收入，则 Amazon Personalize 将按以下方法计算每个用户分数：将用户单击的前 25 个最昂贵建议物品产生的总收入除以用户单击的所有推荐物品产生的总收入。然后，Amazon Personalize 会返回所有用户分数的标准化平均值。`average_rewards_at_k` 越接近 1，表示您期望从建议中获得的每个用户平均收入就越多。

有关更多信息，请参阅 针对其他目标优化解决方案。

normalized_discounted_cumulative_gain_with_event_weights_at_k

当您为具有事件配置的解决方案创建解决方案版本（训练模型）时，Amazon Personalize 会生成 `normalized_discounted_cumulative_gain_with_event_weights_at_k` 指标。`normalized_discounted_cumulative_gain_with_event_weights_at_k` 的分数表示在考虑为每个事件类型设置的事件权重的情况下，解决方案版本的表现情况。

它与 k 阶标准化折损累积增益（NDCG）类似，只不过对每个正确预测的奖励分数会进行加权处理。与之不同的是，在原始 K 阶 NDCG 中，每个正确预测的权重均为 1。例如，如果“购买”权重为 0.3，“点击”权重为 0.1，正确预测“购买”项目将获得 1.5 的奖励分数，而预测“点击”项目将获得 0.5 的奖励分数。

有关更多信息，请参阅 使用事件配置优化解决方案。

趋势预测准确度

如果您使用 Trending-Now 食谱训练解决方案版本，则表示模型推荐的物品的受欢迎程度的增长率。趋势预测准确度越高（越接近 1），模型就越能正确识别热门物品。

为了计算受欢迎程度增长率，Amazon Personalize 将所有推荐物品的受欢迎程度增长率除以前 25 个热门物品的总受欢迎程度增长率。这些物品来自测试集中的实际交互。

根据您的数据分布和您选择的趋势发现频率，趋势预测准确度的值可以为 0.0。

命中率（K 项命中率）

如果您使用 USER_SEGMENTATION 食谱训练解决方案版本，则表示预测的相关度最高的 K 个结果中与实际用户相匹配的平均用户数。实际用户是指与测试集中的物品进行实际交互的用户。K 是最相关的用户中前 1% 的用户。该值越高，表示预测越准确。

查全率（K 项查全率）

如果您使用 USER_SEGMENTATION 食谱训练解决方案版本，则表示预测的相关度最高的 K 个结果中与实际用户相匹配的平均预测用户百分比。实际用户是指与测试集中的物品进行实际交互的用户。K 是最相关的用户中前 1% 的用户。该值越高，表示预测越准确。

查全率

如果您使用 Next-Best-Action配方训练解决方案版本，则该指标会告诉您您的解决方案版本在发现用户将要与之交互的操作方面的表现如何。

为了计算 `recall`，对于数据集中的每个操作，Amazon Personalize 将经正确预测会采取该操作的用户数量除以测试集中实际采取该操作的总用户数量。然后，Amazon Personalize 计算数据集中所有操作的平均值。

例如，如果 100 名用户在测试集中执行了操作，而 Amazon Personalize 预测其中有 50 名用户将采取该操作，则该操作的 `recall` 为 `50 / 100 = .50`。然后，Amazon Personalize 将此计算应用于所有操作并返回平均值。

曲线下方的区域 (AUC)

如果您使用 PERSONALIZED_ACTIONS 配方训练了解决方案版本，则为您的解决方案版本的“接收者操作特征”曲线下方的区域。该指标告诉您，解决方案版本在正确识别用户将要采取的操作方面的表现如何。

“接收者操作特征”曲线绘制了解决方案版本的性能。它绘制了不同阈值下的真阳性（正确预测为相关的操作）和假阳性（错误地预测为相关的操作）的比率。曲线下方的区域 (AUC) 是根据其曲线总结解决方案版本的性能的分数。

解决方案版本的 AUC 可介于 0 和 1 之间。越接近 1，模型就越能更好地预测用户的相关操作。

物品受欢迎程度

如果您使用`aws-semantic-similarity`配方对解决方案版本进行了训练，则该指标会衡量一件商品相对于目录中其他商品的参与度（点赞、观看次数、分享次数等）。物品受欢迎程度分数越高（越接近 1），与其他物品相比，该物品累积的用户互动越多。

为了计算商品的受欢迎程度，Amazon Personalize 会计算与每件商品互动的独立用户总数（通过浏览、点击、购买等事件）。然后，对数据集中所有项目的原始参与度进行标准化，得出的分数介于 0 和 1 之间。分数接近 0 表示参与度低，而接近 1 的分数表示目录中商品的平均参与度高。

物品新鲜度

如果您使用`aws-semantic-similarity`配方对解决方案版本进行了训练，则该指标会衡量某项商品相对于其他商品最近添加到您的目录的时间。物品的新鲜度分数越高（越接近 1），则该物品的创建时间越短。

为了计算商品的新鲜度，Amazon Personalize 使用商品数据集中的 CREATION_TIMESTAMP 来确定每件商品的存放时间。年龄是根据训练时间和项目创建时间戳之间的差额计算的。然后对所有物

品的年龄进行标准化，较新的物品获得的分数接近 1。分数越接近 0 表示旧物品越多，而越接近 1 的分数表示目录中的平均新物品越多。

物品相似度

如果您使用aws-semantic-similarity配方训练了解决方案版本，则此指标会根据其内容、属性和元数据来衡量推荐项目与您的输入项在语义上的相关程度。项目相似度分数越高（越接近 1），推荐的项目就越接近您的参考项目的特征。

为了计算商品的相似度，Amazon Personalize 使用源自商品元数据和属性的语义嵌入来计算每件推荐商品和输入商品之间的相似度分数。分数接近 0 表示平均而言，目录中项目之间的语义相似度较低，而接近 1 的分数表示平均而言，目录中项目之间的语义相似度很高。该指标可帮助您了解模型在推荐中捕捉语义关系的效果如何。

示例

下面是一个可为特定用户生成建议列表的解决方案版本的简单示例。第二个和第五个推荐与此用户的测试数据中的记录匹配。这些是相关的推荐。如果将 K 设置为 5，将为该用户生成以下指标。

reciprocal_rank

计算：1/2

结果：0.5000

normalized_discounted_cumulative_gain_at_5

计算：(1/log(1 + 2) + 1/log(1 + 5)) / (1/log(1 + 1) + 1/log(1 + 2))

结果：0.6241

precision_at_5

计算：2/5

结果：0.4000

其他资源

有关通过测试评估解决方案版本的信息，请参阅使用 A/B A/B 测试来衡量 Amazon Personalize 生成的建议的有效性。要深入了解推荐器系统的不同类型的指标，请参阅以下外部资源：

- MRR vs MAP vs NDCG：排名感知评估指标及何时使用这些指标

- 折扣累计增益：您应该了解的排名指标
- 推荐器系统的 K 项查全率和精度
- 推荐器系统的排名评估指标
- 接收者操作特征

通过市场活动部署 Amazon Personalize 解决方案版本

要获取包含自定义资源的实时建议，完成手动创建解决方案版本后，您就可以通过市场活动部署解决方案版本了。

市场活动 部署的解决方案版本（经过训练的模型）具有预调配的事务容量，用于生成实时建议。创建市场活动后，您可以使用 GetRecommendations 或 GetPersonalizedRanking API 操作来获取建议。如果您获得批量物品推荐或用户细分，则无需创建市场活动。有关更多信息，请参阅获取批量物品推荐或获取批量用户细分。

在创建市场活动时，您可以配置以下内容：

- 您可以将市场活动配置为自动更新以使用解决方案的最新解决方案版本。有关更多信息，请参阅 市场活动自动更新。
- 您可以在推荐中启用物品元数据。有关更多信息，请参阅 推荐中的物品元数据。
- 您可以为该市场活动指定预置的每秒最低事务数。这是 Amazon Personalize 为该市场活动预置的基准事务吞吐量。它会设置市场活动处于活动状态时的最低账单费用。有关更多信息，请参阅 预调配的最低每秒事务数和自动扩缩。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 创建市场活动。如果您想要更改现有市场活动的设置，例如在推荐中启用元数据，则必须更新您的市场活动。有关更多信息，请参阅 更新 Amazon Personalize 市场活动的配置。

当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关市场活动成本的信息，请参阅 Amazon Personalize 定价。

主题

- 市场活动自动更新
- 预调配的最低每秒事务数和自动扩缩
- 推荐中的物品元数据
- 创建市场活动（控制台）

- 创建市场活动 (AWS CLI)
- 创建市场活动 (AWS SDK)

市场活动自动更新

创建市场活动时，您可以启用市场活动自动更新。通过自动更新，市场活动会自动更新以部署解决方案的最新自动或手动训练解决方案版本。这样您就能够更轻松地使市场活动保持最新状态。

例如，如果您的解决方案使用自动训练，每七天创建一个新的解决方案版本，则您的市场活动将自动更新为在每周训练中使用最新的解决方案版本。如果您不使用市场活动自动更新，则必须手动更新市场活动以部署最新的已训练模型。

- 要在使用 Amazon Personalize 控制台创建市场活动时启用市场活动自动更新，请在市场活动详细信息中选择自动更新以使用解决方案的最新解决方案版本。您可以在市场活动详细信息页面上找到最新更新的时间戳。

  有关更多信息，请参阅 创建市场活动（控制台）。
- 要在使用 CreateCampaign API 操作时启用市场活动自动更新，请在 `SolutionVersionArn` 参数中以 `SolutionArn/$LATEST` 格式指定解决方案的 Amazon 资源名称（ARN）。在 `campaignConfig` 中，将 `enableMetadataWithRecommendations` 设置为 `true`。

  要获取最新市场活动更新的时间戳，您可以使用 DescribeCampaign API 操作并查看响应中的 `latestCampaignUpdate` 详细信息。

  有关向您展示如何启用自动更新的代码示例，请参阅创建市场活动 (AWS CLI)或创建市场活动 (AWS SDK)。

预调配的最低每秒事务数和自动扩缩

> **Important**
>
> 高 `minProvisionedTPS` 会增加您的成本。我们建议从 1 开始设置 `minProvisionedTPS`（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 `minProvisionedTPS`。

在创建 Amazon Personalize 市场活动时，您可以为该市场活动指定预置的每秒最低事务数 (`minProvisionedTPS`)。这是 Amazon Personalize 为该市场活动预置的基准事务吞吐量。

它会设置市场活动处于活动状态时的最低账单费用。事务是单个 GetRecommendations 或 GetPersonalizedRanking 请求。默认 minProvisionedTPS 为 1。

如果您的 TPS 增加到超过 minProvisionedTPS，则 Amazon Personalize 会自动纵向扩展和缩减预置容量，但从不会低于 minProvisionedTPS。增加容量时会有短时间延迟，这可能会导致事务丢失。当您的流量减少时，容量将恢复到 minProvisionedTPS。

您需要为预置的最低 TPS 付费，或者，如果您的请求超过 minProvisionedTPS，则需要为实际 TPS 付费。实际 TPS 是您发出的建议请求的总数。我们建议从低 minProvisionedTPS 开始，使用 Amazon CloudWatch 指标跟踪您的使用情况，然后根据需要增加 minProvisionedTPS。

有关市场活动成本的更多信息，请参阅 Amazon Personalize 定价。

推荐中的物品元数据

> **Important**
>
> 如果您使用 User-Personalization-v2 或 Personalized-Ranking-v2 配方，则元数据不会产生额外费用。对于所有其他配方和所有域应用场景，会产生额外费用。有关更多信息，请参阅 Amazon Personalize 定价。

获得推荐后，您可以让 Amazon Personalize 在推荐结果中包括物品元数据。在请求中，可以从物品数据集选择要包括的列。Amazon Personalize 会在建议响应中返回每个物品的这些数据。

您可以使用元数据来丰富用户界面中的建议，例如将电影的类型添加到轮播中。或者，您可以用它来直观地评测建议质量。如果您在应用程序中使用生成式人工智能，则可以将元数据插入人工智能提示中，以生成相关性更高的内容。有关 Amazon Personalize 与生成式人工智能结合使用的更多信息，请参阅Amazon Personalize 和生成式人工智能。

启用元数据

要向建议添加元数据，您必须拥有包含一列元数据的物品数据集。您不一定要在训练中使用元数据。有关创建数据集的信息，请参阅创建架构和数据集。有关更新数据的信息，请参阅训练后更新数据集中的数据。

如果您使用 User-Personalization-v2 或 Personalized-Ranking-v2 配方，则新市场活动会自动选择在推荐结果中包括物品元数据。您没有手动为市场活动启用元数据。对于所有其他配方和域应用场景，必须启用元数据选项：

- 要使用 Amazon Personalize 控制台启用元数据，请在创建市场活动时，在市场活动详细信息中选择在建议结果中返回物品元数据。有关更多信息，请参阅 创建市场活动（控制台）。
- 要使用 AWS SDK 或 AWS CLI 启用元数据，请使用 CreateCampaign API 操作并在 `campaignConfig` 中将 `enableMetadataWithRecommendations` 设置为 `true`。有关更多信息，请参阅创建市场活动 (AWS CLI)或创建市场活动 (AWS SDK)。

创建市场活动（控制台）

> **Important**
>
> 当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关市场活动成本的信息，请参阅 Amazon Personalize 定价。

在您的解决方案版本状态变为“活动”后，您就可以通过 Amazon Personalize 市场活动进行部署了。

创建市场活动（控制台）

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 选择包含要部署的解决方案版本的数据集组。
3. 在导航窗格中，在自定义资源下选择市场活动。
4. 在市场活动选项卡上，选择创建市场活动。
5. 在创建新的市场活动页面上，对于市场活动详细信息，提供以下信息：
   - 市场活动名称 – 输入市场活动的名称。您在此处输入的文本将显示在市场活动控制面板和详细信息页面上。
   - 解决方案 – 选择您刚创建的解决方案。
   - 自动更新以使用解决方案的最新解决方案版本 - 选择此选项可让市场活动自动使用最新的活动解决方案版本。如果不选择此选项，则每次要部署新的解决方案版本时都必须手动更新市场活动。有关更多信息，请参阅 市场活动自动更新。
   - 解决方案版本 ID - 如果不通过市场活动自动更新来使用最新的解决方案版本，请选择要部署的解决方案版本的 ID。
   - 预置的每秒最低事务数（在 API 中称为 minProvisionedTPS）– 设置 Amazon Personalize 支持的预置的每秒最低事务数。较高的值会增加费用。我们建议从 1（默认值）开始设置。使用

Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 minProvisionedTPS。有关更多信息，请参阅 预调配的最低每秒事务数和自动扩缩。

- 在推荐结果中返回物品元数据 - 如果您希望在推荐结果中包括元数据，请选择此选项。如果启用了此选项，则在获取建议时可以指定物品数据集中的列。有关更多信息，请参阅 推荐中的物品元数据。

6. 如果您使用了 User-Personalization 配方，则在市场活动配置中，可以选择输入浏览权重和浏览项目中断时长的值。有关更多信息，请参阅 User-Personalization。
7. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
8. 选择创建市场活动。
9. 在市场活动详细信息页面上，当市场活动状态变为活动 时，您可以使用市场活动来获取建议并记录展示量。有关更多信息，请参阅 从 Amazon Personalize 获取建议。

   当市场活动状态为“活动”时，市场活动即已准备就绪。如果您重新训练解决方案版本，或者想要更改市场活动设置，则必须更新您的市场活动。有关更多信息，请参阅 更新 Amazon Personalize 市场活动的配置。

创建市场活动 (AWS CLI)

> **Important**
>
> 当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关市场活动成本的信息，请参阅 Amazon Personalize 定价。

在您的解决方案版本变为“活动”后，您就可以通过 Amazon Personalize 市场活动进行部署了。要使用 AWS CLI 创建市场活动，请使用 create-campaign 命令。

以下代码示例演示了如何创建市场活动。该示例部署了使用 User-Personalization 配方的解决方案的最新解决方案版本。它创建的市场活动会自动更新以使用未来的解决方案版本。该代码使用以下配置：

- 它将市场活动配置为自动更新，以便使用适用于您的解决方案的最新解决方案版本：solution-version-arn 的格式为 *solution ARN*/$LATEST，syncWithLatestSolutionVersion 为 True。要使用该代码，请将 solution ARN 替换为解决方案的 Amazon 资源名称（ARN）。

  要禁用自动 syncWithLatestSolutionVersion，请仅指定解决方案版本 ARN（不带 /$LATEST），并将 syncWithLatestSolutionVersion 设置为 False。

- 它将 enableMetadataWithRecommendations 选项设置为 True。这样会启用推荐请求选项，以便将物品数据集中的物品元数据与推荐结果一起包括在内。要禁用此选项，请将其设置为 False。有关更多信息，请参阅 推荐中的物品元数据。
- 它将 min-provisioned-tps 设置为 1（默认值）。我们建议从 1 开始设置 minProvisionedTPS（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 minProvisionedTPS。有关更多信息，请参阅 预调配的最低每秒事务数和自动扩缩。

要获得所有参数的完整列表，请参阅CreateCampaign。

```
aws personalize create-campaign \
--name campaign-name \
--solution-version-arn solution-arn/\$LATEST \
--min-provisioned-tps 1 \
--campaign-config "{"\"syncWithLatestSolutionVersion"\": "true",
 "\"enableMetadataWithRecommendations"\": "true"}"
```

当市场活动状态为“活动”时，市场活动即已准备就绪。要获取当前状态，请调用 DescribeCampaign 并检查 status 字段是否为 ACTIVE。

如果您重新训练解决方案版本，但市场活动未自动更新为使用最新的解决方案版本，或者如果您想要更改市场活动设置，则必须更新您的市场活动。有关更多信息，请参阅 更新 Amazon Personalize 市场活动的配置。

Amazon Personalize 为您提供用于管理市场活动的操作（如 ListCampaigns）以列出您创建的市场活动。您可以通过调用 DeleteCampaign 来删除活动。如果您删除一个市场活动，不会删除属于该市场活动的一部分的解决方案版本。

在创建市场活动后，可以使用它来进行推荐。有关更多信息，请参阅 从 Amazon Personalize 获取建议。

创建市场活动 (AWS SDK)

> **Important**
>
> 当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关市场活动成本的信息，请参阅 Amazon Personalize 定价。

在您的解决方案版本变为“活动”后，您就可以通过 Amazon Personalize 市场活动进行部署了。要使用 AWS SDK 创建市场活动，请使用 CreateCampaign API 操作。

以下代码示例演示了如何创建市场活动。该代码部署了使用 User-Personalization 配方的解决方案的最新解决方案版本。它创建的市场活动会自动更新以使用未来的解决方案版本。该代码使用以下配置：

- 它将市场活动配置为自动更新，以便使用适用于您的解决方案的最新解决方案版本：solutionVersionArn 的格式为 *solution ARN*/$LATEST，syncWithLatestSolutionVersion 为 True。要使用该代码，请将 solution ARN 替换为解决方案版本的 Amazon 资源名称（ARN）。

  要禁用自动 syncWithLatestSolutionVersion，请仅指定解决方案版本 ARN（不带 /$LATEST），并将 syncWithLatestSolutionVersion 设置为 False。
- 它将 enableMetadataWithRecommendations 选项设置为 True。这样会启用推荐请求选项，以便将物品数据集中的物品元数据与推荐结果一起包括在内。要禁用此选项，请将其设置为 False。有关更多信息，请参阅 推荐中的物品元数据。
- 它将 minProvisionedTPS 设置为 1（默认值）。对于 minProvisionedTPS，我们建议从 1（默认值）开始设置。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 minProvisionedTPS。有关更多信息，请参阅 预调配的最低每秒事务数和自动扩缩。

要获得所有参数的完整列表，请参阅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();
```

当市场活动状态为“活动”时，市场活动即已准备就绪。要获取当前状态，请调用 DescribeCampaign 并检查 `status` 字段是否为 `ACTIVE`。

如果您手动重新训练解决方案版本，或如果您想要更改市场活动设置，则必须更新您的市场活动。有关更多信息，请参阅 更新 Amazon Personalize 市场活动的配置。

Amazon Personalize 为您提供用于管理市场活动的操作（如 ListCampaigns）以列出您创建的市场活动。您可以通过调用 DeleteCampaign 来删除活动。如果您删除一个市场活动，不会删除属于该市场活动的一部分的解决方案版本。

在创建您的市场活动后，请使用它来进行推荐。有关更多信息，请参阅 从 Amazon Personalize 获取建议。

更新 Amazon Personalize 市场活动的配置

要更改市场活动的 最低预调配的 TPS、手动部署新的解决方案版本，或修改市场活动的配置（例如，打开在推荐中包含元数据的选项），您必须手动更新市场活动。

以下内容不需要手动更新市场活动：

- 如果您的市场活动使用自动市场活动更新，则不必更新市场活动即可部署解决方案的最新自动或手动训练解决方案版本。有关更多信息，请参阅 市场活动自动更新。
- 使用 User-Personalization-v2 User-Personalization Next-Best-Action、或，Amazon Personalize 每两小时自动更新一次您的最新解决方案版本，以便在推荐中加入新的项目或操作。您的市场活动会自动使用更新的解决方案版本。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包手动更新广告活动。

> **Note**
>
> 要更新市场活动以打开在推荐中包含元数据的选项，您必须拥有包含一列元数据的物品数据集。您不一定要在训练中使用元数据。
> 如果您的广告系列之前部署了使用 User-Personalization-v2 或的解决方案版本 Personalized-Ranking-v2，而您正在切换到旧版本的配方，则默认情况下，包含元数据的选项处于关闭状态。您可以在更新市场活动时将其启用。有关更多信息，请参阅 推荐中的物品元数据。

主题

- 更新市场活动（控制台）
- 更新广告活动 (AWS CLI)
- 更新广告活动 (AWS 软件开发工具包)

更新市场活动（控制台）

要部署手动再训练的解决方案版本或更改市场活动配置，您必须更新市场活动。

更新市场活动（控制台）

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择要更新的市场活动所在的数据集组。
3. 在导航窗格中，选择市场活动。
4. 在市场活动页面上，选择要更新的市场活动。
5. 在市场活动详细信息页面上，选择更新。
6. 在更新市场活动页面上，进行更改。例如，如果您要部署经过再训练的解决方案版本，请为解决方案版本 ID 选择新解决方案版本的标识号。
7. 选择更新。Amazon Personalize 会更新市场活动以使用新的解决方案版本和任何更改的配置。

更新广告活动 (AWS CLI)

要部署新的解决方案版本，更改市场活动的最低预调配的 TPS，或更改市场活动配置，您必须更新市场活动。使用以下 `update-campaign` 命令，通过 AWS CLI更新市场活动以使用新的解决方案版本。

将 `campaign arn` 替换为要更新的市场活动的 Amazon 资源名称 (ARN)。将 `new solution version arn` 替换为要部署的解决方案版本。

```
aws personalize update-campaign \
--campaign-arn campaign arn \
--solution-version-arn new solution version arn \
--min-provisioned-tps 1
```

更新广告活动 (AWS 软件开发工具包)

要部署新的解决方案版本，更改市场活动的最低预调配的 TPS，或更改市场活动配置，您必须更新市场活动。使用以下代码使用 适用于 Python (Boto3) 的 AWS SDK 或 SDK for Java 2.x 更新广告系列。要获得参数的完整列表，请参阅 UpdateCampaign。

SDK for Python (Boto3)

使用以下 update_campaign 方法部署新的解决方案版本。将 campaign arn 替换为要更新的市场活动的 Amazon 资源名称 (ARN)，将 new solution version arn 替换为新解决方案版本 ARN，并选择更改 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

使用以下 updateCampaign 方法，更新市场活动以使用新的解决方案版本。将以下内容作为参数传递：Amazon Personalize 服务客户端、新解决方案版本的 Amazon 资源名称 (ARN) 以及最低预调配的 TPS。

```
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);
    }
}
```

从 Amazon Personalize 获取建议

创建推荐器或创建市场活动后，就可以获取建议。根据您的资源，您可以实时获得建议，也可以通过批量工作流程获得建议。

- 使用自定义资源，您可以获得实时建议或批量建议。要获得实时建议，您必须先创建自定义市场活动，然后才能获得建议。要获得批量建议，您无需创建市场活动。
- 对于域数据集组中的推荐器，您只能获得实时建议。

以下主题介绍了如何以及在何时使用每种推荐类型。对于批量推荐和实时推荐，您都可以筛选结果。有关更多信息，请参阅 筛选建议和用户细分。

主题

- 建议分数
- Amazon Personalize 中的实时物品推荐
- Amazon Personalize 中的实时操作推荐
- 获取个性化排名（自定义资源）
- 利用上下文元数据提高建议的相关性
- 使用自定义资源获取批量物品推荐
- 使用自定义资源获取批量用户细分

建议分数

通过使用 User-Personalization-v 2、用户个性化、2、个性化排名和 PERSONALIZED_AC Personalized-Ranking-v TIONS 配方创建的自定义解决方案，Amazon Personalized_ACTIONS 在推荐中包含每件商品的分数。这些分数表示 Amazon Personalize 对于用户接下来将选择哪个物品或操作的相对确定性。分数越高，意味着确定性越大。

- 有关 User-Personalization-v 2 分数和用户个性化的信息，请参阅推荐评分的工作原理（自定义资源）。
- 有关 PERSONALIZED_ACTIONS 配方的分数的信息，请参阅操作建议评分的工作原理。
- 有关 Personalized-Ranking-v 2 分数和个性化排名建议的信息，请参阅。个性化排名评分的工作原理

对于批量推理作业，物品分数的计算方法如 推荐评分的工作原理（自定义资源）和 个性化排名评分的工作原理中所述。您可以在批量推理作业的输出 JSON 文件中查看分数。

Amazon Personalize 中的实时物品推荐

如果您的应用场景或配方生成了物品推荐，那么在创建推荐器或创建市场活动之后，即可为用户获取实时的个性化或相关物品推荐。

如果您的域应用场景或配方提供实时个性化（例如热门精选应用场景或 User-Personalization-v2 配方），则当您记录用户与目录的交互时，Amazon Personalize 会根据用户的最新活动更新推荐。有关记录实时事件和个性化设置的更多信息，请参阅记录实时事件以影响推荐。

获得实时物品推荐后，您可以执行以下操作：

- 如果您将市场活动配置为返回推荐物品的元数据，则可以指定要包括在 GetRecommendations API 操作中的列。或者，您可以在使用 Amazon Personalize 控制台测试市场活动时指定列。有关代码示例，请参阅获取实时推荐的物品元数据。有关为市场活动启用元数据的信息，请参阅推荐中的物品元数据。有关为推荐器启用元数据的信息，请参阅在推荐中为 Amazon Personalize 中的域推荐器启用元数据。
- 对于某些应用场景和配方，您可以在推荐请求中指定推广。推广 定义了其他业务规则，这些规则适用于可配置的建议物品子集。有关更多信息，请参阅 通过实时推荐推广物品。
- 您可以根据自定义条件筛选结果。例如，您可能不想推荐用户已经购买的产品，或者只推荐针对特定年龄段的物品。有关更多信息，请参阅 筛选建议和用户细分。

> **Note**
>
> 如果您使用了 PERSONALIZED_RANKING 自定义食谱，请参阅获取个性化排名（自定义资源）。

主题

- 推荐评分的工作原理（自定义资源）
- User-Personalization-v2 的推荐理由
- 获取实时物品推荐
- 获取实时推荐的物品元数据

- 通过实时推荐推广物品

推荐评分的工作原理（自定义资源）

借助 User-Personalization-v2 和 User-Personalization 配方，Amazon Personalize 可根据用户的交互数据和元数据为物品生成分数。这些分数表示 Amazon Personalize 对于用户是否与接下来的物品交互的相对确定性。分数越高，意味着确定性越大。

> **Note**
>
> Amazon Personalize 不会显示域推荐器或者 Similar-Items、SIMS 或 Popularity-Count 食谱的分数。有关 Personalized-Ranking 建议分数的信息，请参阅 个性化排名评分的工作原理。

Amazon Personalize 会为每个物品生成一个分数，表示该物品与其他物品之间的相对关系，分数范围从 0 到 1（包括 0 和 1）。借助 User-Personalization-v2，Amazon Personalize 可为部分物品生成分数。借助 User-Personalization，Amazon Personalize 可对目录中的所有物品评分。

如果您使用 User-Personalization-v2 并对推荐应用筛选条件，那么根据筛选条件删除的推荐数量，Amazon Personalize 可能会添加占位符物品。这样做是为了符合推荐请求的 `numResults`。这些物品是基于交互数据量的热门物品，符合您的筛选条件。它们没有针对用户的相关性评分。

对于 User-Personalization-v2 和 User-Personalization，所有分数的总和等于 1。例如，如果您要获取用户的影片推荐，并且物品数据集和交互数据集中显示了三部影片，则其分数可能为 `0.6`、`0.3` 和 `0.1`。同样，如果您的库存中有 1 万部电影，则得分最高的电影的分数可能非常小（平均分数为 `.001`），但由于评分是相对的，推荐仍然有效。

在数学术语中，每个用户-物品对（u,i）的分数是根据以下公式计算的，其中 `exp` 是指数函数，$\bar{w}_u$ 和 $w_i/_j$ 分别是用户和物品嵌入内容，希腊字母西格玛（Σ）表示包含分数的所有物品的总和：

$$\text{score}(u,i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_j \exp(\bar{w}_u^\top w_j)}$$

User-Personalization-v2 的推荐理由

如果您使用 User-Personalization-v2，模型通常不会推荐的物品会包含一个 `reason` 清单。这些理由说明了为什么在推荐中包括该物品。可能的原因包括：

- 推广物品 - 表示您在推荐请求中应用的推广活动中包括该物品。
- 浏览 - 表示该物品包括在浏览中。使用浏览时，推荐包括的物品具有更少的交互数据或用户相关性。有关浏览的更多信息，请参阅浏览。
- 热门物品 - 表示该物品作为占位符热门物品包括在内。如果您使用筛选条件，那么根据筛选条件删除的推荐数量，Amazon Personalize 可能会添加占位符物品以满足推荐请求的 `numResults`。这些物品是基于交互数据的热门物品，符合您的筛选条件。它们没有针对用户的相关性评分。

获取实时物品推荐

您可以通过 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 从 Amazon Personalize 推荐器或自定义市场活动中获取实时物品推荐。

主题

- 获取物品建议（控制台）
- 获取物品建议 (AWS CLI)
- 获取物品建议 (AWS SDK)

获取物品建议（控制台）

要通过 Amazon Personalize 控制台获取建议，您需要在推荐器（域数据集组）或自定义市场活动的详情页面上提供请求信息。

获得建议

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 选择包含您要使用的市场活动或推荐器的数据集组。
3. 在导航窗格中，选择市场活动 或推荐器。
4. 选择目标市场活动或推荐器。
5. 对于活动，在测试市场活动结果下，根据您使用的食谱输入建议请求详细信息。对于推荐器，选择测试推荐器，然后根据使用案例输入建议请求详细信息。

   如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 `sessionId`（就像是其 `userId`），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。

6. （可选）选择筛选器。有关更多信息，请参阅 筛选建议和用户细分。
7. 如果您使用上下文元数据，请为每个上下文提供数据。对于每个上下文，为键 输入元数据字段。对于值，输入上下文数据。有关更多信息，请参阅 利用上下文元数据提高建议的相关性。
8. 如果您在市场活动或推荐器的建议中启用了元数据，请为物品数据集列选择要包含在建议结果中的元数据列。有关为市场活动启用元数据的信息，请参阅推荐中的物品元数据。有关为推荐器启用元数据的信息，请参阅在推荐中为 Amazon Personalize 中的域推荐器启用元数据。
9. 如果您想推广一部分物品，可以选择填写推广字段。有关更多信息，请参阅 通过实时推荐推广物品。
10. 选择获得建议。此时，将显示一个包含针对用户推荐的前 25 个物品的表格。如果您使用 User-Personalization-v2，则每个推荐物品都会包含一份理由清单，说明为什么在推荐中包括该物品。有关更多信息，请参阅 User-Personalization-v2 的推荐理由。

获取物品建议 (AWS CLI)

使用以下代码从市场活动中获得建议。要从推荐器中获得建议，请将 `campaign-arn` 参数替换为 `recommender-arn`。

指定要获取建议的用户的 ID 及市场活动或推荐器的 Amazon 资源名称 (ARN)。这时，将显示为用户推荐的前 10 个物品的列表。如果您使用 User-Personalization-v2，则每个推荐物品都会包含一份理由清单，说明为什么在推荐中包括该物品。有关更多信息，请参阅 User-Personalization-v2 的推荐理由。

要更改推荐物品的数量，请更改 `numResults` 的值。默认值为 25 个物品。最大值为 500 个物品。如果您使用 RELATED_ITEMS 食谱来训练支持市场活动的解决方案版本，请将 `user-id` 参数替换为 `item-id` 并指定物品 ID。

如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 `sessionId`（就像是其 `userId`），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

获取物品建议 (AWS SDK)

以下代码展示了如何使用 AWS SDK 从市场活动中为用户获取 Amazon Personalize 推荐。要从推荐器中获得建议，请将 `campaignArn` 参数替换为 `recommenderArn`。

指定要获取建议的用户的 ID 及市场活动或推荐器的 Amazon 资源名称 (ARN)。这时，将显示为用户推荐的前 10 个物品的列表。如果您使用 User-Personalization-v2，则每个推荐物品都会包含一份理由清单，说明为什么在推荐中包括该物品。有关更多信息，请参阅 User-Personalization-v2 的推荐理由。

要更改推荐物品的数量，请更改 numResults 的值。默认值为 25 个物品。最大值为 500 个物品。如果您使用 RELATED_ITEMS 食谱来训练支持市场活动的解决方案版本，请将 userId 参数替换为 itemId 并指定物品 ID。

如果您在市场活动或推荐器的建议中启用了元数据，则可以指定要包含在响应中的物品数据集元数据列。有关代码示例，请参阅在推荐中包括物品元数据（AWS SDK）。有关启用元数据的信息，请参阅推荐中的物品元数据。

如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 sessionId（就像是其 userId），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。

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();
```

获取实时推荐的物品元数据

如果您将市场活动或推荐器配置为返回推荐物品的元数据，则可以指定要包括在 GetRecommendations API 操作中的列。或者，您可以在使用 Amazon Personalize 控制台测试市场活动时指定列。

有关为市场活动启用元数据的信息，请参阅推荐中的物品元数据。有关为推荐器启用元数据的信息，请参阅在推荐中为 Amazon Personalize 中的域推荐器启用元数据。

以下代码示例展示了如何使用 AWS CLI 或 AWS SDK 指定要包括的元数据列。要使用 Amazon Personalize 控制台完成此操作，请在测试市场活动时指定列。有关更多信息，请参阅 获取物品建议（控制台）。

主题

- 在推荐中包括物品元数据（AWS CLI）
- 在推荐中包括物品元数据（AWS SDK）

在推荐中包括物品元数据（AWS CLI）

如果您在市场活动或推荐器的建议中启用了元数据，则可以指定要包含在响应中的物品数据集元数据列。以下代码示例显示了如何在请求建议时指定元数据列。

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10 \
--metadata-columns "{\"ITEMS\": [\"\"columnNameA\"\",\"\"columnNameB\"\"]}"
```

在推荐中包括物品元数据（AWS SDK）

如果您在市场活动或推荐器的建议中启用了元数据，则可以指定要包含在响应中的物品数据集元数据列。以下代码示例显示了如何在请求建议时指定元数据列。

```
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'])
```

通过实时推荐推广物品

对于所有域应用场景和一些自定义配方，您可以在获得实时推荐时指定推广。

推广 定义了其他业务规则，这些规则适用于可配置的建议物品子集。例如，您可能有一款流应用程序，想要推广自己的节目和电影，但也想推荐相关的内容。您可以使用推广来指定一定比例的推荐物品必须来自内部 类别。根据您的食谱和任何请求筛选器，其余的推荐物品将继续是相关建议。

要应用推广，请在建议请求中指定以下内容：

- 要应用推广筛选器的推荐物品的百分比。
- 指定推广标准的筛选器。有关更多信息，请参阅 推广筛选器。

在建议响应中，推广物品相对于其他推荐物品是随机放置的，但相对于其他推广物品按排序顺序列出。根据您的食谱，不属于推广的推荐物品将按与用户的相关性、受欢迎程度或相似度进行排序。如果没有足够的物品符合推广标准，则结果将包含尽可能多的推广物品。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包对推荐进行促销。

主题

- 支持推广的使用案例和食谱

- 推广筛选器
- 推广新物品
- 推广物品（控制台）
- 促销商品 (AWS CLI)
- 促销商品 (AWS 软件开发工具包)

支持推广的使用案例和食谱

所有使用案例都支持推广。以下自定义食谱支持推广：

- User-Personalization-v2 和 User-Personalization 配方
- Similar-Items 和 SIMS 配方
- Trending-Now 和 热门程度-计数 配方

推广筛选器

将推广应用于建议请求时，您可以选择一个指定推广标准的筛选器。您可以使用现有筛选器或创建新的筛选器。您可以像在 Amazon Personalize 中创建和管理其他筛选器一样创建和管理推广筛选器。有关创建和管理筛选器的信息，请参阅筛选结果。

推广筛选器和您在推广之外选择的筛选器（请求筛选器）之间的唯一区别是 Amazon Personalize 如何应用它们。推广筛选器仅适用于推广物品，而请求筛选器仅适用于剩余的推荐物品。如果您指定了请求筛选器和推广筛选器，并且想要将这两个筛选器应用于推广物品，则推广筛选器的表达式必须同时包含这两个表达式。合并两个表达式的方式取决于您使用的数据集。有关筛选器表达式、其规则以及如何创建此类内容的更多信息，请参阅筛选条件表达式。

筛选器表达式示例

以下表达式仅包含“内部”类别中的物品。如果您想在建议中推广自己的内容，则可以使用此表达方式。

```
INCLUDE ItemID WHERE Items.OWNER IN ("in-house")
```

以下表达式仅包括在您指定的时间戳之后创建的物品。您可以使用此表达式来推广推荐中的新物品。

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

以下表达式显示了如何将请求筛选器应用于推广物品。它仅包括作为推广物品的可用服装。在这种情况下，还会在请求筛选器表达式中使用 `Items.AVAILABLE IN ("True")`，因此所有建议都适用于可用的物品。

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") AND Items.AVAILABLE IN ("True")
```

有关筛选器示例的更完整列表，请参阅筛选器表达式示例。

推广新物品

如果您使用 User-Personalization-v2 配方，Amazon Personalize 会向用户推荐最相关的物品，并更频繁地推荐带有交互数据的现有物品。为确保推荐中包括一些新物品，您可以根据创建时间戳对包括物品的推荐请求应用推广。

如果您尚未使用推广，则您的筛选条件表达式可以推广在特定日期之后创建的物品：

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

如果您已经在使用推广，则可以创建一个将推广和新物品条件语句关联起来的表达式：

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") OR Items.CREATION_TIMESTAMP > $DATE
```

推广物品（控制台）

要使用 Amazon Personalize 控制台在建议中推广某些商品，请创建筛选器，然后在建议请求中提供推广详情。有关其他字段的信息，请参阅获取物品建议（控制台）。

在建议中推广物品

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择包含您要使用的市场活动或推荐器的数据集组。
3. 创建一个筛选器来指定推广条件（如果尚未创建）。为推广创建筛选器的方式与创建请求筛选器的方法相同。有关创建和管理筛选器的信息，请参阅筛选结果。
4. 在导航窗格中，选择推荐器 或活动。
5. 选择目标市场活动或推荐器。
6. 对于市场活动，在测试市场活动结果下，根据您使用的食谱输入建议请求详细信息。对于推荐器，选择测试推荐器，然后输入建议请求详细信息。

7. （可选）为请求选择筛选器。此筛选器仅适用于非推广物品。有关创建和管理筛选器的信息，请参阅筛选结果。
8. 如果您使用上下文元数据，请为每个上下文提供数据。对于每个上下文，为键 输入元数据字段。对于值，输入上下文数据。有关更多信息，请参阅 利用上下文元数据提高建议的相关性。
9. 对于推广，指定以下内容：
   - 推广物品百分比：输入要应用推广的推荐物品所占的百分比。
   - 筛选器：选择用于指定推广标准的筛选器。此筛选器适用于推广物品，而不是您在步骤 7 中可能指定的任何请求筛选器。
   - 筛选器参数：如果您的推广使用的筛选器带有占位符参数，请为每个参数输入值以设置筛选条件。要为一个参数使用多个值，请用英文逗号分隔每个值。
10. 选择获得建议。此时，将显示一个包含针对用户推荐的前 25 个物品的表格。推广物品列显示该物品是否因为您的推广而被收录。推广物品相对于其他推荐物品是随机放置的，但相对于其他推广物品按排序顺序列出。根据您的使用案例或食谱，不属于推广的推荐物品将按与用户的相关性、受欢迎程度或相似度进行排序。如果没有足够的物品符合推广标准，则结果将包含尽可能多的推广物品。

促销商品 (AWS CLI)

以下代码展示了如何使用 AWS CLI 和自定义广告系列来推广推荐中的商品。要使用推荐器推广物品，请将 `campaign-arn` 参数替换为 `recommender-arn`，并为推荐器指定 Amazon 资源名称 (ARN)。对于推广字段，指定以下内容：

- 名称：为推广命名。建议响应使用该名称来标识推广的物品。
- percent-promoted-items：要应用推广的推荐物品所占的百分比。在此示例中，50% 的物品将成为推广物品。
- filterArn：指定用于定义推广条件的筛选器的 Amazon 资源名称 (ARN)。有关更多信息，请参阅 推广筛选器。
- 参数名称和值：如果您的筛选器表达式有任何参数，请提供参数名称（区分大小写）和值。例如，如果您的筛选器表达式具有 $GENRE 参数，请提供 GENRE 作为键，并提供一个或多个类型（例如 Comedy）作为值。用英文逗号分隔多个值。使用时 AWS CLI，对于每个值，都必须使用/字符对引号和/字符进行转义。以下代码示例显示如何设置值的格式。

该代码显示了如何同时使用请求筛选器和推广筛选器。推广筛选器仅适用于推广物品，而请求筛选器仅适用于剩余的推荐物品。有关更多信息，请参阅 推广筛选器。

有关其他字段的信息，请参阅获取物品建议 (AWS SDK) 和使用上下文元数据获得个性化排名。

```
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\\\"\"}
}]"
```

这时，将显示推荐的物品列表。推广物品相对于其他推荐物品是随机放置的，但相对于其他推广物品按排序顺序列出。根据您的食谱，不属于推广的推荐物品将按与用户的相关性、受欢迎程度或相似度进行排序。如果没有足够的物品符合推广标准，则结果将包含尽可能多的推广物品。

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

促销商品 (AWS 软件开发工具包)

以下代码展示了如何使用和 SDK for Java 2.x 适用于 Python (Boto3) 的 AWS SDK 以及自定义广告系列来推广推荐中的商品。要使用推荐器推广物品，请将 `campaignArn` 参数替换为 `recommenderArn`，并为推荐器指定 Amazon 资源名称 (ARN)。对于推广字段，指定以下内容：

- 名称：指定推广的名称。建议响应包含该名称来标识推广的物品。
- 百分比PromotedItems：要应用促销的推荐商品的百分比。
- promotionFilterARN：指定用于定义推广条件的筛选器的 Amazon 资源名称 (ARN)。有关更多信息，请参阅 推广筛选器。
- 任何参数名称和值：如果您的筛选器表达式包含任何参数，请为筛选器表达式中的每个参数提供参数名称（区分大小写）和值。例如，如果您的筛选器表达式具有 $GENRE 参数，请提供 "GENRE" 作为键，并提供一个或多个类型（例如 "\"Comedy"\"）作为值。用英文逗号分隔多个值。例如 `"\"comedy\",\"drama\",\"horror"\"`。

以下代码显示了如何同时使用请求筛选器和推广筛选器。推广筛选器仅适用于推广物品，而请求筛选器仅适用于剩余的推荐物品。有关更多信息，请参阅 推广筛选器。

有关其他字段的信息，请参阅获取物品建议 (AWS SDK) 和使用上下文元数据获得个性化排名。

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();
```

这时，将显示推荐的物品列表。推广物品相对于其他推荐物品是随机放置的，但相对于其他推广物品按排序顺序列出。根据您的食谱，不属于推广的推荐物品将按与用户的相关性、受欢迎程度或相似度进行排序。如果没有足够的物品符合推广标准，则结果将包含尽可能多的推广物品。

```
{
  "itemList": [
      {
          "itemId1": "123",
          "score": .0117211,
          "promotionName": "promotionName"
      },
      {
         "itemId2": "456",
         "score": .0077976
      },
      {
         "itemId3": "789",
         "score": .0067171
      },
      .....
 ]
```

Amazon Personalize 中的实时操作推荐

如果您使用 PERSONALIZED_ACTIONS 配方，则可以实时从市场活动中获得操作建议。您可以使用 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWS SDK 获得操作建议。

主题

- 操作建议评分的工作原理
- 获取操作建议（控制台）
- 获取操作建议 (AWS CLI)
- 获取操作建议 (AWS SDK)

操作建议评分的工作原理

借助 Next-Best-Action 配方，Amazon Personalize 会根据用户将与操作交互的可能性为操作生成分数。分数可以介于 0 - 1.0 之间。越接近 1.0，用户将与操作交互的可能性就越大。

如果您尚未导入任何操作交互数据，则所有建议操作的分数都将为 0.0。如果 Amazon Personalize 在浏览过程中建议操作，则该物品的分数将为 0.0。Amazon Personalize 使用浏览来建议没有操作交互数据的操作。有关浏览的更多信息，请参阅探索。

获取操作建议（控制台）

要通过 Amazon Personalize 控制台获取建议，您需要在自定义市场活动的详情页面上提供请求信息。

获取操作建议

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 选择包含要使用的市场活动的数据集组。
3. 在导航窗格中，在自定义资源下选择市场活动。
4. 选择目标市场活动。
5. 在测试市场活动结果下，输入您的建议请求详细信息。

   如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 `sessionId`（就像是其 `userId`），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。
6. （可选）选择筛选器。有关更多信息，请参阅 筛选建议和用户细分。
7. 选择获得建议。此时将显示一个表，其中包含向用户建议的前 5 个操作。

获取操作建议 (AWS CLI)

使用以下代码从市场活动中获得操作建议。指定要获取其建议的用户的 ID 及市场活动的 Amazon 资源名称 (ARN)。

要更改建议操作的数量，请更改 `numResults` 的值。默认为 5 个操作。最多为 100 个操作。

要按自定义条件筛选操作建议，您可以创建筛选条件并将其应用于 `get-action-recommendations` 操作。有关更多信息，请参阅 筛选建议和用户细分。

如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 sessionId（就像是其 userId），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。

```
aws personalize-runtime get-action-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

获取操作建议 (AWS SDK)

以下代码展示了如何从市场活动中为用户获取 Amazon Personalize 建议。指定要获取其建议的用户的 ID 及市场活动的 Amazon 资源名称 (ARN)。

要更改建议操作的数量，请更改 numResults 的值。默认为 5 个操作。最多为 100 个操作。

要按自定义条件筛选操作建议，您可以创建筛选条件并将其应用于 GetActionRecommendations API 请求。有关更多信息，请参阅 筛选建议和用户细分。

如果您在用户登录（匿名用户）之前为其记录了事件，则可以通过提供来自这些事件的 sessionId（就像是其 userId），获取针对此用户的建议。有关为匿名用户记录事件的更多信息，请参阅为匿名用户记录事件。

```
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'])
```

获取个性化排名（自定义资源）

个性化排名是为特定用户重新排名的推荐物品的列表。要获得个性化排名，请调用 GetPersonalizedRanking API 操作或在控制台中从市场活动中获取建议。

> **Note**
>
> 必须已使用 PERSONALIZED_RANKING 类型的配方创建支持市场活动的解决方案。有关更多信息，请参阅 选择食谱。

主题

- 个性化排名评分的工作原理
- 获取个性化排名（控制台）
- 获取个性化排名 (AWS CLI)
- 获取个性化排名 (AWS SDK)
- Personalized-Ranking 示例笔记本

个性化排名评分的工作原理

与使用 User-Personalization-v2 和 User-Personalization 配方创建的解决方案的 `GetRecommendations` 操作返回的分数相似，`GetPersonalizedRanking` 分数总和为 1，但只有输入物品获得分数且推荐分数往往更高。如果某个项目在最近一次训练中不存在，则该项目的分数为 0。

在数学上，GetPersonalizedRanking 的评分函数与 `GetRecommendations` 相同，但它只考虑输入物品。这意味着分数接近 1 的可能性更高，因为将分数进行划分的其他选择更少：

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_{j \in \text{input}} \exp(\bar{w}_u^\top w_j)}$$

获取个性化排名（控制台）

要从 Amazon Personalize 控制台获取用户的个性化排名，请选择您要使用的市场活动，然后提供他们的用户 ID，指定您要为该用户排名的物品列表，（可选）选择筛选器，并（可选）提供任何上下文数据。

获得针对用户的个性化排名

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。

2. 选择包含要使用的市场活动的数据集组。
3. 在导航窗格中，选择市场活动。
4. 在活动页面上，选择目标市场活动。
5. 在测试市场活动结果 下，输入要为其获取建议的用户的用户 ID。
6. 对于物品 ID，输入要为用户排名的物品列表。
7. （可选）选择筛选器。有关更多信息，请参阅 筛选建议和用户细分。
8. 如果您在市场活动的建议中启用了元数据，请为物品数据集列选择要包含在建议结果中的元数据列。有关启用元数据的信息，请参阅推荐中的物品元数据。
9. 如果活动使用上下文元数据（有关要求，请参阅利用上下文元数据提高建议的相关性），则可以选择提供上下文数据。

   对于每个上下文，为键 输入元数据字段，对于值，输入上下文数据。
10. 选择获取个性化物品排名。此时将显示一个表格，其中包含按用户预期兴趣顺序排名的物品。

获取个性化排名 (AWS CLI)

以下代码示例显示了如何使用 AWS CLI 获得个性化排名的不同变体。

主题

- 获取个性化排名
- 在个性化排名中包含物品元数据

获取个性化排名

使用以下 `get-personalized-ranking` 命令，通过 AWS CLI 获得个性化排名。指定市场活动的 Amazon 资源名称 (ARN)、用户的用户 ID，并提供要为用户排名的物品的物品 ID 列表（每个项用空格隔开）。要排名的物品必须位于您用于训练解决方案版本的数据中。这将显示已排名的建议列表。Amazon Personalize 会将列表中的第一个物品视为用户最感兴趣的物品。

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
--input-list 3 4 10 8 12 7
```

在个性化排名中包含物品元数据

如果您在市场活动的建议中启用了元数据，则可以指定要包含在响应中的物品数据集元数据列。有关启用元数据的信息，请参阅推荐中的物品元数据。

以下代码示例显示了如何在请求个性化排名时指定元数据列。

```
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\"\"]}"
```

获取个性化排名 (AWS SDK)

以下代码示例显示了如何使用 AWS SDK 获得个性化排名的不同变体。

主题

- 获取个性化排名
- 在个性化排名中包含物品元数据
- 使用上下文元数据获得个性化排名

获取个性化排名

以下代码展示了如何为用户获取个性化排名。指定用户的 ID 和要为用户排名的物品 ID 列表。物品 ID 必须位于您用于训练解决方案版本的数据中。这将返回已排名的建议列表。Amazon Personalize 会将列表中的第一个物品视为用户最感兴趣的物品。

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();
```

在个性化排名中包含物品元数据

如果您在市场活动的建议中启用了元数据，则可以指定要包含在响应中的物品数据集元数据列。有关启用元数据的信息，请参阅推荐中的物品元数据。

以下代码示例显示了如何在请求个性化排名时指定元数据列。

```
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'])
```

使用上下文元数据获得个性化排名

使用以下代码，根据上下文元数据获取个性化排名。对于 `context`，对于每个键值对，提供元数据字段作为键，提供上下文数据作为值。在以下示例代码中，键为 `DEVICE`，值为 `mobile phone`。替换这些值，将 `Campaign ARN` 和 `User ID` 替换为您自己的值。此外，将 `inputList` 更改为用于训练解决方案的数据中的物品 ID 列表。Amazon Personalize 会将列表中的第一个物品视为用户最感兴趣的物品。

```
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'])
```

Personalized-Ranking 示例笔记本

有关展示如何使用 Personalized-Ranking 食谱的示例 Jupyter 笔记本，请参阅 Personalize Ranking 示例。

利用上下文元数据提高建议的相关性

要提高建议相关性，请在获得物品建议或获得个性化排名时添加用户的上下文元数据，例如他们的设备类型或一天中的时间。

要使用上下文元数据，项目交互数据集的架构必须具有上下文数据的元数据字段。例如，DEVICE 字段（请参阅为 Amazon Personalize 架构创建架构 JSON 文件）。在考虑用于上下文元数据的字段时，请选择项目交互数据集中历史数据可用值的字段，以及何时获得项目推荐或获得个性化项目排名的字段。

对于域数据集组，以下推荐器使用案例可以使用上下文元数据：

- 为您推荐（ECOMMERCE 域）
- 热门精选（VIDEO_ON_DEMAND 域）

对于自定义资源，使用上下文元数据的食谱包括以下内容：

- User-Personalization-v2 和 User-Personalization
- Personalized-Ranking-v2 和 Personalized-Ranking

有关情境信息的更多信息，请参阅以下 M AWS achine Learning 博客文章：利用情境信息提高 Amazon Personalize 推荐的相关性。

您可以通过 Amazon Personalize 控制台、AWS Command Line Interface (AWS CLI) 或 AWSSDKs，获取包含上下文元数据的推荐。

使用情境相关元数据获取推荐 (AWSPython SDK)

要提高建议相关性，请在获得物品建议或获得个性化排名时添加用户的上下文元数据，例如他们的设备类型或一天中的时间。

使用以下代码，根据上下文元数据获取建议。对于 `context`，对于每个键值对，提供元数据字段作为键，提供上下文数据作为值。在以下示例代码中，键为 `DEVICE`，值为 `mobile phone`。替换这些值，将 `Campaign ARN` 和 `User ID` 替换为您自己的值。如果您创建了推荐器，请将 `campaignArn` 替换为 `recommenderArn`。这时，将显示为用户推荐的物品列表。

```
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'])
```

使用自定义资源获取批量物品推荐

借助自定义资源，您可以通过异步批量流获得物品建议。例如，您可以获取电子邮件列表中所有用户的产品建议，或者获得库存中的物品间相似度。

要获取物品的批量建议，您可以使用批量推理作业。批量推理作业 是一种工具，可从 Amazon S3 存储桶导入您的批量输入数据，使用您的自定义解决方案版本生成物品建议，然后将物品建议导出到 Amazon S3 存储桶。根据食谱，您的输入数据是用户或物品列表，或者是用户列表，每个用户列表都含有一组物品。

如果您的解决方案使用 Similar Items 配方，并且您的物品数据集包含文本数据和物品标题数据，则可以为每组物品生成带有主题的批量建议。有关更多信息，请参阅 内容生成器中带有主题的批量建议。

创建自定义解决方案版本后，新数据如何影响批量物品推荐取决于其类型、导入方法和所用的自定义配方。有关新数据如何影响批量推荐的信息，请参阅新数据如何影响批量推荐（自定义资源）。

主题

- 批量工作流程
- 指南和要求
- 批量工作流程评分
- 内容生成器中带有主题的批量建议
- 为批量建议准备输入数据
- 创建批量推理作业
- 批量推理作业输出示例

批量工作流程

批量工作流程如下所示：

1. 准备好您的输入数据并以 JSON 格式将其上传到 Amazon S3 存储桶。输入数据的格式取决于您使用的食谱。请参阅为批量建议准备输入数据。
2. 为输出数据创建一个单独的位置，可以是文件夹，也可以是其他 Amazon S3 存储桶。
3. 创建批量推理作业。请参阅创建批量推理作业。
4. 批量推理完成后，从 Amazon S3 中的输出位置检索物品建议。

指南和要求

以下是获取批量建议的指南和要求：

- 您的 Amazon Personalize IAM 服务角色必须有权读取文件并将其添加到您的 Amazon S3 存储桶。有关授予权限的信息，请参阅批量工作流程的服务角色策略。有关存储桶权限的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的用户策略示例。如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
- 在创建批量推理作业之前，必须创建自定义解决方案和解决方案版本。不过，您无需创建 Amazon Personalize 市场活动。如果您创建了域数据集组，则仍然可以创建自定义资源。
- 要生成带有建议的主题，您必须使用 Similar-Items 配方。而且您必须有一个包含文本数据和物品标题数据的物品数据集。有关带有主题的建议的更多信息，请参阅内容生成器中带有主题的批量建议。
- 输入数据的格式必须按为用户细分准备输入数据所述来设置。
- 您无法通过 Trending-Now 或 Next-Best-Action 配方获得批量建议。
- 如果您使用带有占位符参数的筛选器，则必须在 `filterValues` 对象的输入数据中包含参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。
- 我们建议您使用与输入数据不同的位置来存储输出数据（文件夹或其他 Amazon S3 存储桶）。
- 批量推荐可能与实时推荐不完全相同。这是因为批量推理作业需要更长的时间才能完成，并且仅考虑作业开始前 15 分钟可用的数据。

批量工作流程评分

批量建议包括以下分数：

- 对于 User-Personalization 和 Personalized-Ranking 食谱，Amazon Personalize 计算批量推理作业建议分数，如 推荐评分的工作原理（自定义资源）和 个性化排名评分的工作原理所述。您可以在批量推理作业的输出 JSON 文件中查看分数。
- 使用 Similar-Items 配方，如果您获得带有主题的批量建议，Amazon Personalize 会根据主题与每个物品的相关程度对每组相关物品进行排名。每个物品都包含从 0 到 1 的分数。分数越高，该物品与主题的关系就越密切。有关带有主题的建议的更多信息，请参阅内容生成器中带有主题的批量建议。

内容生成器中带有主题的批量建议

> **Important**
>
> 当您获得带有主题的批量建议时，会产生额外的成本。有关更多信息，请参阅 Amazon Personalize 定价。

如果您使用 Similar-Items 食谱，Amazon Personalize 内容生成器可以向批量建议添加描述性主题。内容生成器是一种生成式人工智能功能，由 Amazon Personalize 管理。

当您获得带有主题的批量建议时，Amazon Personalize 内容生成器会为每组相似的物品添加一个描述性主题。该主题基于物品数据集中的物品描述和物品名称数据。Amazon Personalize 将主题包含在批量推理作业的输出中。您可以使用这些主题使应用程序或营销信息中的文本更具吸引力。

例如，如果您收到关于早餐食品的相关物品建议，Amazon Personalize 可能会生成一个主题，例如快起床吧或早晨必备品。您可以使用该主题来替换通用的轮播标题，例如经常一起购买。或者，您可以将该主题纳入促销电子邮件或营销活动中，以获得新的菜单选项。

AWS 不监视内容生成器中的主题。要确认主题质量，您可以使用为每个建议的物品生成的分数。有关更多信息，请参阅 对带有主题的批量建议进行排名和评分。

主题

- 支持的区域
- 指南和要求
- 对带有主题的批量建议进行排名和评分
- 生成带有主题的批量建议

支持的区域

Amazon Personalize 内容生成器仅在以下 AWS 地区可用：

- 美国东部（弗吉尼亚州北部）
- 美国西部（俄勒冈）
- 亚太地区（东京）

指南和要求

以下是生成带有主题的建议的指南和要求：

- 您的输入文件最多可以有 100 个物品。有关批量建议的输入数据的信息，请参阅为批量建议准备输入数据。
- 您的解决方案必须使用Similar-Items 食谱。
- 您必须拥有包含以下数据的物品数据集。这些数据可以帮助生成更相关的主题。
  - 它必须有一个文本字段，例如 DESCRIPTION 字段。有关文本数据的信息，请参阅非结构化文本元数据。
  - 它必须有一个包含物品名称数据的字符串列，例如 TITLE 字段。

  如果您的物品数据集没有此数据，则可以添加它。有关更新现有数据的信息，请参阅训练后更新数据集中的数据。

对带有主题的批量建议进行排名和评分

当您收到带有主题的批量建议时，Amazon Personalize 会根据主题与每个物品的相关性程度对每组物品进行排名。每个物品都包含一个大致范围为 -0.1 和 0.6 的分数。分数越高，该物品与主题的关系就越密切。您可以使用分数来设置一个阈值，以仅显示与主题密切相关的物品。

例如，Amazon Personalize 返回的主题可能为 `For your sweet tooth`，相关物品及其分数可能是：硬糖（分数为 0.19884521）、巧克力（分数为 0.17664525）、苹果（分数为 0.08994528）、冰棒（分数为 0.14294521）、红薯（分数为 0.07794527）和胡萝卜（分数为 0.04994523）。在您的应用程序中，您可以添加一条规则，仅包含分数等于 `.10` 或更高的物品，从而去掉水果和蔬菜。

以下示例显示生成带有主题的电影建议的批量推理作业的输出格式。

```
{"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,-
...
```

生成带有主题的批量建议

要生成带有主题的批量建议，请按照批量工作流程中所述完成批量工作流程。您以与 RELATED_ITEMS 配方相同的方式准备输入数据。有关示例，请参阅RELATED_ITEMS 配方。

创建批量推理作业时，您可以启用主题生成并指定物品数据集的物品标题列。

- 有关使用 Amazon Personalize 控制台创建用于生成主题的批量推理作业的信息，请参阅创建批量推理作业。
- 有关演示如何使用创建生成主题的 适用于 Python (Boto3) 的 AWS SDK 批量推理作业的代码示例，请参阅创建生成主题的批量推理作业。

为批量建议准备输入数据

批量推理作业可从 Amazon S3 存储桶导入您的批量输入 JSON 数据，使用您的自定义解决方案版本生成建议，然后将物品建议导出到 Amazon S3 存储桶。在获取批量建议之前，您必须准备好 JSON 文件并将其上传到 Amazon S3 存储桶。我们建议您在 Amazon S3 存储桶中创建输出文件夹，或者使用单独的输出 Amazon S3 存储桶。然后，您可以使用相同的输入数据位置运行多个批量推理作业。

如果您使用带有占位符参数的筛选器（如 $GENRE），则必须在输入 JSON 的 filterValues 对象中提供参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。

准备和导入数据

1. 根据您的食谱设置批量输入数据的格式。您无法通过 Trending-Now 食谱获得批量建议。

   - 对于 USER_PERSONALIZATION 食谱和 Popularity-Count 食谱，您的输入数据是包含 userId 列表的 JSON 文件
   - 对于 RELATED_ITEMS 食谱，您的输入数据是 itemID 列表
   - 对于 PERSONALIZED_RANKING 食谱，您的输入数据是 userId 列表，每个用户 ID 都与一组 itemId 配对

用新行分隔每行。有关输入数据的示例，请参阅批量推理作业输入和输出 JSON 示例。

2. 将您的输入 JSON 上传到 Amazon S3 存储桶中的输入文件夹。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
3. 为输出数据创建一个单独的位置，可以是文件夹，也可以是其他 Amazon S3 存储桶。通过为输出 JSON 创建单独的位置，您可以使用相同的输入数据位置运行多个批量推理作业。
4. 创建批量推理作业。Amazon Personalize 会将解决方案版本的建议输出到输出数据位置。

批量推理作业输入和输出 JSON 示例

如何将输入数据格式设置为您使用的食谱的格式。如果您使用带有占位符参数的筛选器（如 `$GENRE`），则必须在输入 JSON 的 `filterValues` 对象中提供参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。

以下各节列出了格式正确的 JSON 批量推理作业输入和输出示例。您无法通过 Trending-Now 食谱获得批量建议。

主题

- USER_PERSONALIZATION 食谱
- POPULAR_ITEMS 配方（仅 Popularity-Count）
- PERSONALIZED_RANKING 食谱
- RELATED_ITEMS 配方

USER_PERSONALIZATION 食谱

下面显示了 USER_PERSONALIZATION 配方的格式正确的 JSON 输入和输出示例。如果您使用 User-Personalization-v2，则每个推荐物品都会包含一份理由清单，说明为什么在推荐中包括该物品。此列表可以为空。有关可能原因的信息，请参阅User-Personalization-v2 的推荐理由。

Input

用新行分隔每个 `userId`，如下所示。

```
{"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.
...
```

POPULAR_ITEMS 配方（仅 Popularity-Count）

下面显示了 Popularity-Count 配方的格式正确的 JSON 输入和输出示例。您无法通过 Trending-Now 食谱获得批量建议。

Input

用新行分隔每个 `userId`，如下所示。

```
{"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"]}}
...
```

PERSONALIZED_RANKING 食谱

以下显示了 PERSONALIZED_RANKING 配方的格式正确的 JSON 输入和输出示例。

Input

用新行分隔每个 userId 和要排名的 itemIds 列表，如下所示。

```
{"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]}}
...
```

RELATED_ITEMS 配方

下面显示了 RELATED_ITEMS 配方的格式正确的 JSON 输入和输出示例。

Input

用新行分隔每个 itemId，如下所示。

```
{"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"]}}
...
```

下面显示了带有主题的 Similar-Items 配方的格式正确的 JSON 输入和输出示例。

Input

用新行分隔每个 itemId，如下所示。

```
{"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
...
```

创建批量推理作业

创建批量推理作业，根据 Amazon S3 的输入数据获取针对用户的批量物品建议。输入数据可以是 JSON 格式的用户列表或物品列表（或两者兼而有之）。您可以使用 Amazon Personalize 控制台 AWS Command Line Interface (AWS CLI) 或 AWS SDKs创建批量推理作业。

创建批量推理作业时，您可以指定输入和输出位置的 Amazon S3 路径。Amazon S3 基于前缀。如果您为输入数据位置提供前缀，则 Amazon Personalize 会使用与该前缀匹配的所有文件作为输入数据。例如，如果您提供 s3://amzn-s3-demo-bucket/folderName 且存储桶中还有一个路径为 s3://amzn-s3-demo-bucket/folderName_test 的文件夹，则 Amazon Personalize 会使用两个文件夹中的所有文件作为输入数据。要仅使用特定文件夹中的文件作为输入数据，请使用前缀分隔符作为 Amazon S3 路径的结尾，例如 /: s3://amzn-s3-demo-bucket/folderName/。有关 Amazon S3 如何组织对象的更多信息，请参阅组织、列出和处理您的对象。

有关 Amazon Personalize 中批量工作流程的更多信息（包括权限要求、建议评分以及准备和导入输入数据），请参阅使用自定义资源获取批量物品推荐。

主题

- 创建批量推理作业（控制台）

- 创建批量推理作业 (AWS CLI)
- 创建批量推理作业 (AWS SDKs)

创建批量推理作业（控制台）

完成为批量建议准备输入数据后，就可以创建批量推理作业了。此过程假定您已创建了一个解决方案和一个解决方案版本（经过训练的模型）。

创建批量推理作业（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 在导航窗格的自定义资源下，选择批量推理作业。
4. 选择创建批量推理作业。
5. 选择批量推理作业类型。
   - 要生成不带主题的物品建议，请选择物品建议。
   - 如果您使用 Similar-Items 配方并想向相似物品组添加描述性主题，请选择内容生成器中带有主题的建议。要生成主题，必须有一个包含物品名称数据和文本数据的物品数据集。有关更多信息，请参阅 内容生成器中带有主题的批量建议。
6. 在批量推理作业详细信息的批量推理作业名称中，指定批量推理作业的名称。
7. 对于解决方案，选择解决方案，然后选择要用于生成建议的解决方案版本 ID。
8. 对于结果数量，可以选择为每行输入数据指定建议的数量。默认值为 25。
9. 如果您的批处理作业生成了带有主题的建议，请在带有主题的建议详细信息中，选择物品数据集中包含物品名称或标题的列。这些数据可以帮助生成更相关的主题。有关更多信息，请参阅 内容生成器中带有主题的批量建议。
10. 在输入源中，指定输入文件的 Amazon S3 路径。

    使用以下语法：**`s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json`**

    您的输入数据必须采用与您的解决方案使用的食谱对应的正确格式。有关输入数据的示例，请参阅批量推理作业输入和输出 JSON 示例。

11. 对于解密密钥，如果您使用自己的密 AWS KMS 钥进行存储桶加密，请指定密钥的 Amazon 资源名称 (ARN)。Amazon Personalize 必须拥有使用您的密钥的权限 有关授予权限的信息，请参阅授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
12. 在输出目标中，指定输出位置的路径。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。

    使用以下语法：**s3://amzn-s3-demo-bucket/<output folder name>/**
13. 对于加密密钥，如果您使用自己的 AWS KMS 密钥进行加密，请指定密钥的 ARN。Amazon Personalize 必须拥有使用您的密钥的权限 有关授予权限的信息，请参阅授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
14. 对于 IAM 服务角色，选择您在设置过程中为 Amazon Personalize 创建的 IAM 服务角色。此角色必须分别对您的输入和输出 Amazon S3 存储桶具有读写权限。
15. 在筛选条件中，（可选）选择一个筛选条件，以将筛选条件应用于批量建议。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。
16. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
17. 选择创建批量推理作业。此时将开始创建批量推理作业，批量推理作业页面将显示，并显示批量推理作业详细信息部分。

    当批量推理作业的状态更改为活动 时，您可以从指定的输出 Amazon S3 存储桶中检索作业的输出。输出文件的名称将采用 *input-name*.out 格式。

创建批量推理作业 (AWS CLI)

完成为批量建议准备输入数据后，就可以通过 CreateBatchInferenceJob 操作创建批量推理作业了。

主题

- 创建批量推理作业
- 创建生成主题的批量推理作业

创建批量推理作业

您可以使用 `create-batch-inference-job` 命令创建批量推理作业。指定作业名称，将 `Solution version ARN` 替换为解决方案版本的 Amazon 资源名称 (ARN)，然后将 `IAM service`

role ARN 替换为您在设置期间为 Amazon Personalize 创建的 IAM 服务角色的 ARN。此角色必须分别对您的输入和输出 Amazon S3 存储桶具有读写权限。（可选）提供筛选器 ARN 来筛选建议。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅筛选批量建议和用户细分（自定义资源）。

将 S3 input path 和 S3 output path 替换为指向您的输入文件和输出位置的 Amazon S3 路径。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。对输入和输出位置使用以下语法：**s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json** 和 **s3://amzn-s3-demo-bucket/<output folder name>/**。

该示例包括可选的 User-Personalization 食谱特定的 itemExplorationConfig 超参数：explorationWeight 和 explorationItemAgeCutOff。（可选）包括 explorationWeight 和 explorationItemAgeCutOff 值以配置浏览。有关更多信息，请参阅 User-Personalization 食谱。

```
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\"}}"
```

创建生成主题的批量推理作业

要为相似的物品生成主题，您必须使用 Similar-Items 配方，并且您的物品数据集必须有一个文本字段和一个物品名称数据列。有关带有主题的建议的更多信息，请参阅内容生成器中带有主题的批量建议。

以下代码创建了一个批量推理作业，该作业可生成带有主题的建议。将 batch-inference-job-mode 保留设置为 THEME_GENERATION。将 COLUMN_NAME 替换为存储物品名称数据的列的名称。

```
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\"}}"
```

创建批量推理作业 (AWS SDKs)

完成为批量建议准备输入数据后，就可以通过 CreateBatchInferenceJob 操作创建批量推理作业了。

主题

- 创建批量推理作业
- 创建生成主题的批量推理作业

创建批量推理作业

可以使用以下代码创建批量推理作业。指定作业名称、解决方案版本的 Amazon 资源名称 (ARN)，以及您在设置期间为 Amazon Personalize 创建的 IAM 服务角色的 ARN。此角色必须对您的输入和输出 Amazon S3 存储桶具有读写权限。

我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。对输入和输出位置使用以下语法：**s3:/amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json** 和 **s3://amzn-s3-demo-bucket/<output folder name>/**。

对于 numResults，指定您希望 Amazon Personalize 为每行输入数据预测的物品数量。（可选）提供筛选器 ARN 来筛选建议。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅 筛选批量建议和用户细分（自定义资源）。

SDK for Python (Boto3)

> 该示例包括可选的 User-Personalization 食谱特定的 itemExplorationConfig 超参数：explorationWeight 和 explorationItemAgeCutOff。（可选）包括 explorationWeight 和 explorationItemAgeCutOff 值以配置浏览。有关更多信息，请参阅 User-Personalization 食谱。

```
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

该示例包括可选的 User-Personalization 食谱特定的 `itemExplorationConfig` 字段：`explorationWeight` 和 `explorationItemAgeCutOff`。（可选）包括 `explorationWeight` 和 `explorationItemAgeCutOff` 值以配置浏览。有关更多信息，请参阅 User-Personalization 食谱。

```
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();
```

处理批处理作业可能需要一段时间才能完成。您可以通过调用 DescribeBatchInferenceJob 和传递 batchRecommendationsJobArn 作为输入参数来检查作业的状态。您也可以通过调ListBatchInferenceJobs用列出您 AWS 环境中的所有 Amazon Personalize 批量推理作业。

创建生成主题的批量推理作业

要为相似的物品生成主题，您必须使用 Similar-Items 配方，并且您的物品数据集必须有一个文本字段和一个物品名称数据列。有关带有主题的建议的更多信息，请参阅内容生成器中带有主题的批量建议。

以下代码创建了一个批量推理作业，该作业可生成带有主题的建议。将 batchInferenceJobMode 保留设置为 "THEME_GENERATION"。将 COLUMNN_NAME 替换为存储物品名称数据的列的名称。

```
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>/"}}
)
```

批量推理作业输出示例

当您创建批量推理作业时，该作业会从 Amazon S3 存储桶导入您的批量输入数据，使用您的解决方案版本生成物品建议，并将建议以 JSON 格式导出到 Amazon S3 存储桶。

以下各节按食谱类型列出了批量推理作业的输出文件示例。您无法通过 Trending-Now 或 Next-Best-Action 配方获得批量建议。

主题

- USER_PERSONALIZATION 食谱
- POPULAR_ITEMS 食谱
- PERSONALIZED_RANKING 食谱
- RELATED_ITEMS 配方

USER_PERSONALIZATION 食谱

下面是 USER_PERSONALIZATION 食谱的输出 JSON 文件示例。

```
{"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
...
```

POPULAR_ITEMS 食谱

以下示例显示了 Popularity-Count 食谱的输出 JSON 文件的格式。您无法通过 Trending-Now 食谱获得批量建议。

```
{"input": {"userId": "12"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "105"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "41"}, "output": {"recommendedItems": ["105", "106", "441"]}}
...
```

PERSONALIZED_RANKING 食谱

以下示例显示了 PERSONALIZED_RANKING 食谱的输出 JSON 文件的格式。

```
{"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]}}
...
```

RELATED_ITEMS 配方

以下示例显示了 RELATED_ITEMS 食谱的输出 JSON 文件的格式。

```
{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}
...
```

以下示例显示了带有主题的 Similar-Items 配方的输出 JSON 文件的格式。有关带有主题的建议的更多信息，请参阅内容生成器中带有主题的批量建议。


```
{"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,-
...
```


使用自定义资源获取批量用户细分

要获取用户细分，您可以使用批量细分作业。批量细分作业 是一种工具，可从 Amazon S3 存储桶导入您的批量输入数据，并使用通过 USER_SEGMENTATION 食谱训练的解决方案版本为每一行输入数据生成用户细分。

根据食谱，输入数据是 JSON 格式的物品列表或物品元数据属性列表。对于物品属性，您的输入数据可以包含基于多个元数据属性创建用户细分的表达式。批量细分作业将用户细分导出到输出 Amazon S3 存储桶。根据每个用户与您输入数据中物品交互的概率，按降序对每个用户细分进行排序。

在生成用户细分时，Amazon Personalize 会考虑来自批量和单独导入的数据集中的数据：

- 对于批量数据，Amazon Personalize 仅使用上次完整解决方案版本训练中提供的批量数据生成细分。而且，它仅使用您以 FULL（替换现有数据）导入模式导入的批量数据。
- 对于单独数据导入操作的数据，Amazon Personalize 使用上次完整解决方案版本训练中提供的数据生成用户细分。要让较新的记录影响用户细分，请创建新解决方案版本，然后创建批量细分作业。

生成用户细分的工作原理如下：

1. 准备好您的输入数据并以 JSON 格式将其上传到 Amazon S3 存储桶。输入数据的格式取决于您使用的食谱和要创建的作业。请参阅为用户细分准备输入数据。
2. 为输出数据创建一个单独的位置，可以是不同的文件夹，也可以是不同的 Amazon S3 存储桶。
3. 创建批量细分作业。请参阅使用批量细分作业来获取用户细分。
4. 批量细分作业完成后，从 Amazon S3 的输出位置检索用户细分。

主题

- 获取用户细分的准则和要求
- 为用户细分准备输入数据
- 使用批量细分作业来获取用户细分
- 批量细分作业输出格式示例

获取用户细分的准则和要求

以下是获取批量细分的指南和要求：

- 您必须使用 USER_SEGMENTATION 食谱。
- 您的 Amazon Personalize IAM 服务角色需要权限来读取文件并将其添加到您的 Amazon S3 存储桶。有关授予权限的信息，请参阅批量工作流程的服务角色策略。有关存储桶权限的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的用户策略示例。

  如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
- 在创建批量推理作业之前，必须创建自定义解决方案和解决方案版本。不过，您无需创建 Amazon Personalize 市场活动。如果您创建了域数据集组，则仍然可以创建自定义资源。
- 输入数据的格式必须按为用户细分准备输入数据所述来设置。
- 如果您使用 Item-Attribute-Affinity 配方，则输入数据中的属性不能包含非结构化文本物品元数据，例如产品描述。
- 如果您使用带有占位符参数的筛选器，则必须在 `filterValues` 对象的输入数据中包含参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。
- 我们建议您使用与输入数据不同的位置来存储输出数据（文件夹或其他 Amazon S3 存储桶）。

为用户细分准备输入数据

批量细分作业使用解决方案版本，根据您在输入 JSON 文件中提供的数据进行用户细分。在获取用户细分之前，您必须准备好 JSON 文件并将其上传到 Amazon S3 存储桶。我们建议您在 Amazon S3 存储桶中创建输出文件夹，或者使用单独的输出 Amazon S3 存储桶。然后，您可以使用相同的输入数据位置运行多个批量推理作业。

如果您使用带有占位符参数的筛选器（如 $GENRE），则必须在输入 JSON 的 filterValues 对象中提供参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。

准备和导入数据

1. 根据您的解决方案使用的食谱，设置批量输入数据的格式。用新行分隔输入数据元素。您的输入数据是 itemIds 列表 (Item-Affinity) 或物品属性列表 (Item-Attribute-Affinity)。
   - 对于物品属性，输入数据可以包括带有 AND 运算符的逻辑表达式，用于为每个查询获取多个物品或属性的用户。有关更多信息，请参阅 为 Item-Attribute-Affinity 配方指定物品属性。
   - 对于物品属性，使用 \ 字符对输入数据中的任何特殊字符以及单引号或双引号进行转义。
   - 有关这两个配方的输入数据示例，请参阅批量细分作业输入和输出 JSON 示例。
2. 将您的输入 JSON 上传到 Amazon S3 存储桶中的输入文件夹。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
3. 为输出数据创建一个单独的位置，可以是文件夹，也可以是其他 Amazon S3 存储桶。通过为输出 JSON 创建单独的位置，您可以使用相同的输入数据位置运行多个批量细分作业。

准备好输入数据并将其上传到 Amazon S3 存储桶后，即可使用批量细分作业生成用户细分。有关更多信息，请参阅 使用批量细分作业来获取用户细分。

主题

- 为 Item-Attribute-Affinity 配方指定物品属性
- 批量细分作业输入和输出 JSON 示例

为 Item-Attribute-Affinity 配方指定物品属性

如果您使用 Item-Attribute-Affinity 配方，则您的输入数据是物品属性列表。您可以混用不同列的元数据。例如，一行可能是数字列，下一行可能是类别列。不能使用非结构化文本物品元数据作为物品属性。

您的输入物品元数据可以包含带有 AND 运算符的逻辑表达式，用于获取多个属性的用户细分。例如，一行输入数据可能是 `{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}` 或 `{"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.audience = "\teen\""}`。

当您将两个属性与 AND 运算符组合在一起时，您会根据用户交互历史记录创建用户细分，其中包含更有可能与同时具有这两个属性的物品进行交互的用户。与筛选器表达式（使用 IN 运算符表示字符串相等）不同，批量细分输入表达式仅支持字符串匹配的相等 = 符号。

批量细分作业输入和输出 JSON 示例

对于批量细分作业，您的输入数据必须是 itemIds 列表（Item-Affinity 食谱）或物品属性列表 (Item-Attribute-Affinity)。每行输入数据都是一个单独的推理查询。根据每个用户与您库存物品交互的概率，按降序对每个用户细分进行排序。

如果您使用带有占位符参数的筛选器（如 $GENRE），则必须在输入 JSON 的 filterValues 对象中提供参数的值。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。

以下是按食谱组织的批量细分作业格式正确的 JSON 输入和输出示例。

Item-Affinity

Input

您的输入数据最多可有 500 个物品。用新行分隔每个 itemId，如下所示。

```
{"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

您的输入数据最多可以有 10 个查询，其中每个查询都是一个或多个非文本物品属性。用新行分隔每个属性或属性表达式，如下所示。

```
{"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"]}}
...
```

使用批量细分作业来获取用户细分

如果您使用了 USER_SEGMENTATION 食谱，则可以创建批量细分作业，以获取解决方案版本的用户细分。根据每个用户与您库存物品交互的概率，按降序对每个用户细分进行排序。根据食谱，您的输入数据必须是 JSON 格式的物品列表（Item-Affinity 食谱）或物品属性（Item-Attribute-Affinity 食谱）。您可以使用 Amazon Personalize 控制台 AWS Command Line Interface （AWS CLI）或创建批量细分任务 AWS SDKs。

创建批量细分作业时，您可以指定输入和输出位置的 Amazon S3 路径。Amazon S3 基于前缀。如果您为输入数据位置提供前缀，则 Amazon Personalize 会使用与该前缀匹配的所有文件作为输入数据。例如，如果您提供 `s3://amzn-s3-demo-bucket/folderName` 且存储桶中还有一个路径为 `s3://amzn-s3-demo-bucket/folderName_test` 的文件夹，则 Amazon Personalize 会使用两个文件夹中的所有文件作为输入数据。要仅使用特定文件夹中的文件作为输入数据，请使用前缀分隔符作为 Amazon S3 路径的结尾，例如 /: `s3://amzn-s3-demo-bucket/folderName/`。有关 Amazon S3 如何组织对象的更多信息，请参阅组织、列出和处理您的对象。

主题

- 创建批量细分作业（控制台）
- 创建批量细分作业 (AWS CLI)
- 创建批量细分作业 (AWS SDKs)

创建批量细分作业（控制台）

完成为批量建议准备输入数据后，就可以创建批量细分作业了。此过程假定您已通过 USER_SEGEMENTATION 食谱，创建了一个解决方案和一个解决方案版本（经过训练的模型）。

创建批量细分作业（控制台）

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 在数据集组页面上，选择您的数据集组。
3. 在导航窗格中选择批量细分作业，然后选择创建批量细分作业。
4. 在批量细分作业详细信息 中，对于批量细分作业名称，为您的批量细分作业指定一个名称。
5. 对于解决方案，选择解决方案，然后选择要用于生成建议的解决方案版本 ID。只有使用了 USER_SEGEMENTATION 食谱，您才能创建批量细分作业。
6. 对于用户数量，可以选择指定 Amazon Personalize 为每个用户细分生成的用户数量。默认值为 25。最大值为 500 万。
7. 对于输入源，指定输入文件的 Amazon S3 路径，或者使用浏览 S3 选择您的 Amazon S3 存储桶。

   使用以下语法：**s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json**

   您的输入数据必须采用与您的解决方案使用的食谱对应的正确格式。有关输入数据的示例，请参阅批量细分作业输入和输出 JSON 示例。
8. 对于输出目标，指定输出位置的路径，或使用浏览 S3 选择您的 Amazon S3 存储桶。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。

   使用以下语法：**s3://amzn-s3-demo-bucket/<output folder name>/**
9. 对于 IAM 角色，选择下列一个选项：

   - 选择创建和使用新的服务角色，然后输入服务角色名称以创建新角色，或者
   - 如果您已经创建了具有正确权限的角色，请选择使用现有服务角色，并选择 IAM 角色。

   您使用的角色必须分别对您的输入和输出 Amazon S3 存储桶具有读写权限。

10. 对于筛选器配置，可以选择一个过滤器，将筛选器应用于用户细分。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。
11. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
12. 选择创建批量细分作业。此时将开始创建批量细分作业，批量细分作业 页面将显示，并显示批量细分作业详细信息 部分。
13. 当批量细分作业的状态更改为活动 时，您可以从指定的输出 Amazon S3 存储桶中检索作业的输出。输出文件的名称将采用 *input-name*.out 格式。

创建批量细分作业 (AWS CLI)

完成为批量建议准备输入数据后，就可以使用以下 create-batch-segment-job 代码创建批量细分作业了。指定作业名称，将 Solution version ARN 替换为解决方案版本的 Amazon 资源名称 (ARN)，然后将 IAM service role ARN 替换为您在设置期间为 Amazon Personalize 创建的 IAM 服务角色的 ARN。此角色必须分别对您的输入和输出 Amazon S3 存储桶具有读写权限。对于 num-results，指定您希望 Amazon Personalize 为每行输入数据预测的用户数量。默认值为 25。最大值为 500 万。（可选）提供 filter-arn 来筛选用户细分。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅 筛选批量建议和用户细分（自定义资源）。

将 S3 input path 和 S3 output path 替换为指向您的输入文件和输出位置的 Amazon S3 路径。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。对输入和输出位置使用以下语法：**s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json** 和 **s3://amzn-s3-demo-bucket/<output folder name>/**。

```
aws personalize create-batch-segment-job \
                --job-name Job name \
                --solution-version-arn Solution version ARN \
                --num-results The number of predicted users \
                --filter-arn Filter ARN \
                --job-input s3DataSource={path=s3://S3 input path} \
                --job-output s3DataDestination={path=s3://S3 output path} \
                --role-arn IAM service role ARN
{
   "batchSegmentJobArn": "arn:aws:personalize:us-west-2:acct-id:batch-segment-job/
batchSegmentJobName"
}
```

创建批量细分作业 (AWS SDKs)

完成为批量建议准备输入数据后，就可以通过 CreateBatchSegmentJob 操作创建批量细分作业了。以下代码展示了如何创建批量细分作业。为作业命名，指定要使用的解决方案版本的 Amazon 资源名称 (ARN)，为您的 Amazon Personalize IAM 角色指定 ARN，并指定您的输入文件和输出位置的 Amazon S3 路径。您的 IAM 服务角色必须分别对您的输入和输出 Amazon S3 存储桶具有读写权限。

我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。对输入和输出位置使用以下语法：**s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>.json** 和 **s3://amzn-s3-demo-bucket/<output folder name>/**。

对于 numResults，指定您希望 Amazon Personalize 为每行输入数据预测的用户数量。默认值为 25。最大值为 500 万。（可选）提供 filterArn 来筛选用户细分。如果您的筛选器使用占位符参数，请确保这些参数的值包含在您的输入 JSON 中。有关更多信息，请参阅 筛选批量建议和用户细分（自定义资源）。

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();
```

处理批处理作业可能需要一段时间才能完成。您可以通过调用 DescribeBatchSegmentJob 和传递 batchSegmentJobArn 作为输入参数来检查作业的状态。您也可以致电列出您 AWS 环境中的所有 Amazon Personalize 批量细分任务ListBatchSegmentJobs。

批量细分作业输出格式示例

批量细分作业可从 Amazon S3 存储桶导入您的批量输入数据，使用通过 USER_SEGMENTATION 食谱训练的解决方案版本生成用户细分，并将细分导出到 Amazon S3 存储桶。

以下各节列出了按配方进行的批量细分作业的 JSON 输出示例。

主题

- Item-Affinity
- Item-Attribute-Affinity

Item-Affinity

以下示例显示了 Item-Affinity 食谱输出 JSON 文件的格式。

```
{"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

以下示例显示了 Item-Attribute-Affinity 食谱的输出 JSON 文件的格式。

```
{"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"]}}
...
```

筛选建议和用户细分

通过域推荐器或自定义市场活动获取建议时，您可以根据自定义条件筛选结果。例如，您可能不想推荐用户已经购买的产品，或者只推荐针对特定年龄段的物品。

同样，对于 USER_SEGMENTATION 食谱，您可能不希望在用户细分中包含某些类型的用户。通过筛选结果，您可以控制向用户推荐的物品或将包含在用户细分中的用户。

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 和 AWS 软件开发工具包创建、编辑、删除和应用筛选条件。

- 对于实时推荐，您可以在调用、或 GetPersonalizedRanking 操作时应用筛选条件并指定任何筛选参数值。 GetRecommendations GetActionRecommendations您也可以当在控制台中获得来自市场活动或推荐器的建议时应用筛选条件。

  当您获得包含个性化或相关物品配方或使用案例的实时物品建议时，您可以在请求中指定促销活动。推广 使用筛选器来定义其他业务规则，这些规则适用于可配置的建议物品子集。有关更多信息，请参阅 通过实时推荐推广物品。
- 对于批量工作流程，您可以在输入 JSON 中包含任何筛选器参数值。然后，在创建批量推理作业或批量细分作业时，指定筛选器的 Amazon 资源名称 (ARN)。有关更多信息，请参阅 筛选批量建议和用户细分（自定义资源）。

新记录的筛选器更新

对于您使用 PutEvents 或 PutActionInteractions 操作导入的数据，Amazon Personalize 会在导入后的几秒钟内用新数据更新数据集组中的所有筛选条件。例如，如果您的筛选条件从推荐中移除已购买的商品，并且您为使用该 PutEvents 操作的用户记录了购买事件，则该项目将在记录该事件后的几秒钟内从该用户的 future 推荐中删除。

对于批量或单独导入的所有其它数据，Amazon Personalize 会在上次导入后的 20 分钟内用新数据更新数据集组中的任何筛选条件。

主题

- 筛选条件表达式
- 筛选实时建议
- 筛选批量建议和用户细分（自定义资源）

筛选条件表达式

要配置筛选器，您必须使用格式正确的筛选器表达式。筛选器表达式由 `dataset.field` 格式的数据集和字段标识符以及逻辑运算符、关键字和值组成。对于值，您可以指定固定值或添加占位符参数，以便设置获得建议时的筛选器。

您可以使用筛选条件表达式根据以下数据集中的数据筛选推荐中的物品、用户或操作：

- 物品交互：您可以使用筛选条件表达式，根据交互数据来包括或排除物品或用户。例如，您可以排除用户已经点击的项目（用于项目推荐），或者仅包括对项目进行评分的用户（针对 Item-Affinity 食谱）。对于所有配方类型，您都只能根据事件类型进行筛选。您无法根据其他交互元数据（例如上下文元数据）进行筛选。您无法对 Item-Attribute-Affinity 食谱 使用物品交互筛选条件。

  Amazon Personalize 考虑每位用户每种事件类型最多 100 次最新交互。这是一个可调节的限额。您可以使用服务配额控制台请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。
- 操作交互：使用筛选条件表达式，根据事件类型来包括或排除用户交互过的操作。例如，您可以排除用户已经执行的操作。您无法根据其他操作交互元数据进行筛选。

  Amazon Personalize 考虑每位用户每种事件类型最多 300 次最新操作交互。这是一个可调节的限额。您可以使用服务配额控制台请求增加配额。
- 物品：使用筛选器表达式根据特定物品条件包含或排除物品。您不能使用筛选器根据非结构化文本物品元数据（例如产品描述）来包含或排除物品。如果您的域用例或自定义配方生成了相关的项目推荐，例如 Similar-Items 食谱或 M ore Like X 域用例，则您可以使用筛选表达式根据您在推荐请求中指定的项目的属性来包含或排除项目。
- 用户：对于物品和操作建议，如果您有用户数据集，则可以根据 `CurrentUser` 排除或包含物品或操作。对于个性化建议、热门物品和操作建议，这是您要获得其建议的用户。对于相关物品，这是一个您可以在建议请求中指定的可选用户。

  对于用户细分，您可以使用筛选条件表达式，根据用户的属性（例如 `Users.MEMBERSHIP_STATUS`）在用户细分中包含或排除用户。
- 操作：使用筛选条件表达式根据特定操作条件包含或排除操作。Amazon Personalize 会自动根据 `Action expiration timestamp` 和 `Repeat frequency` 数据排除操作。您无法创建其它基于此数据进行筛选的自定义筛选条件。

有关筛选器表达式元素的完整列表，请参阅筛选器表达式元素。有关筛选器表达式的示例，请参阅筛选器表达式示例。

主题

- 指南和要求
- 筛选条件表达式结构和元素
- 筛选器表达式示例

指南和要求

创建筛选条件表达式时，请注意以下指南和要求：

- 您不能使用筛选器根据非结构化文本物品元数据（例如产品描述）来包含或排除物品。
- 如果您基于物品或操作交互数据进行筛选，则只能根据事件类型进行筛选。您无法根据其他交互元数据（例如上下文元数据）进行筛选。
- Amazon Personalize 仅在匹配事件类型时忽略大小写。
- 您不能在一个表达式中使用物品交互数据集和物品数据集。要创建按交互和物品数据集（或相反）进行筛选的过滤器，必须将两个或多个表达式链接在一起。有关更多信息，请参阅 组合多个表达式。
- 您不能在一个表达式中使用物品交互数据集和操作数据集。要创建先按物品交互数据集、后按操作数据集（或相反）进行筛选的筛选条件，必须将两个或更多表达式链接在一起。有关更多信息，请参阅 组合多个表达式。
- 您无法对 Item-Attribute-Affinity 食谱 使用物品交互筛选条件。
- 您无法创建使用架构中布尔型值进行筛选的筛选器表达式。要根据布尔值进行筛选，请使用字段类型为字符串 的架构，并在数据中使用值 `"True"` 和 `"False"`。或者，您可以使用类型 int 或 long 以及值 `0` 和 `1`。
- 一个表达式或链接在一起的多个表达式中筛选器的不同数据集字段的最大数量为 10。一个数据集组中所有筛选器不同数据集字段的最大数量为 20。
- 只有当您的域用例或自定义配方生成相关项目推荐（例如配方或 M ore Like X 域用例）时，您才能对 CurrentItem 元素应用过滤器。 Similar-Items
- 不能在使用 NOT_IN 运算符的筛选器表达式中使用占位符参数。可以改用 IN 运算符并使用相反的操作。例如，使用 Include，而不是 Exclude（反之亦然）。
- 您无法创建基于 `Action expiration timestamp` 和 `Repeat frequency` 数据进行筛选的筛选条件。Amazon Personalize 会根据这些数据自动筛选操作建议。

筛选条件表达式结构和元素

本部分包含有关筛选条件表达式的结构及其元素的信息。

主题

- 筛选条件表达式结构
- 筛选器表达式元素

筛选条件表达式结构

筛选器表达式的一般结构如下所示：

```
EXCLUDE/INCLUDE ItemID/ActionID/UserID WHERE dataset type.field IN/NOT IN (value/
parameter)
```

您可以手动创建筛选器表达式，也可以使用控制台中的表达式生成器获取有关表达式语法和结构的帮助。

筛选器表达式元素

使用以下元素创建筛选器表达式：

INCLUDE 或 EXCLUDE

使用 INCLUDE，将建议限制为仅符合筛选条件的物品，或者 使用 EXCLUDE，删除所有符合筛选条件的物品。

ItemID/ActionID/UserID

在 INCLUDE 或 EXCLUDE 元素之后使用上述一个元素。您使用的元素取决于您是在筛选物品（针对物品建议）、筛选操作（针对操作建议）还是筛选用户（针对用户细分）。

WHERE

使用 WHERE 检查物品、操作或用户的条件。您必须在 ItemID、ActionID 或 UserID 之后使用 WHERE 元素。

AND/OR

要在同一个筛选器表达式中将多个条件链接在一起，请使用 AND 或 OR。使用 AND 或 OR 将条件链接在一起，只能影响第一个条件中使用的数据集字段。

Dataset.field

提供要按 dataset.field 格式筛选建议的数据集和元数据字段。例如，要根据项目数据集中的流派字段筛选项目推荐，可以在筛选表达式 Items.genres 中使用。

IF 条件

仅 使用 IF 条件检查 CurrentUser 的条件，并且只能在表达式的末尾使用一次。但是，您可以使用 AND 扩展 IF 条件。

CurrentUser.attribute

要仅 在 IF 条件下根据获得建议的用户筛选物品建议，请使用 CurrentUser 并提供用户字段。例如 CurrentUser.AGE。

CurrentItem.attribute

对于仅相关物品食谱和使用案例，使用 CurrentItem.attribute，根据您在请求相关物品建议时指定的物品属性筛选物品。例如，CurrentItem.GENRE 或 CurrentItem.PRICE。

只有当您的域用例或自定义配方生成相关项目推荐（例如配方或 M ore Like X 域用例）时，您才能对 CurrentItem 元素应用过滤器。 Similar-Items 首次使用 CurrentItem 元素创建筛选器时，可能需要几分钟才能创建筛选器。如果您使用 AWS KMS 加密，则创建筛选器最多可能需要 15 分钟。

IN/NOT 在

使用 IN 或 NOT IN 作为比较运算符，根据匹配（或不匹配）一个或多个字符串值进行筛选。Amazon Personalize 仅根据精确的字符串进行筛选。

比较运算符

使用 =、<、<=、>、>= 和 != 运算符来测试数值数据（包括在占位符参数中传递的数据）是否相等。

星号 (*) 字符

使用 * 包含或排除所有类型的交互。仅 将 * 用于使用 Interactions 数据集 EVENT_TYPE 字段的筛选器表达式。

竖线分隔符

使用竖线分隔符 (|) 将多个表达式链接在一起。有关更多信息，请参阅 组合多个表达式。

参数

对于使用比较运算符或 IN 运算符的表达式，使用美元符号 ($) 和参数名称，将占位符参数添加为值。例如 $GENRES。在此示例中，获得建议时，您需提供一个或多个要筛选的类型。

> **Note**
>
> 当您将参数名称添加到表达式时，可以定义该名称。参数名称不必与字段名称匹配。我们建议您使用与字段名称相似且容易记住的参数名称。在对建议请求应用筛选器时，您可以使用参数名称（区分大小写）。有关演示如何在使用 AWS SDK 时使用占位符参数应用过滤器的示例，请参阅。应用过滤器 (AWS 软件开发工具包)

筛选器表达式示例

使用以下部分中的筛选器表达式来了解如何构建自己的筛选器表达式。

主题

- 物品建议筛选条件表达式示例
- 用户细分筛选器表达式
- 操作建议筛选条件表达式示例
- 组合多个表达式

物品建议筛选条件表达式示例

以下筛选条件表达式显示了如何根据物品交互、物品元数据和用户元数据筛选物品建议。它们按数据类型组织。

主题

- 物品交互数据
- 物品数据
- 用户数据

物品交互数据

以下表达式根据事件类型（例如单击）或您使用 $EVENT_TYPE 参数获取建议时指定的事件类型，排除物品。

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

以下表达式排除用户单击或流式传输过的物品。

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click", "stream")
```

以下表达式仅包含用户已单击的物品。

```
INCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click")
```

物品数据

以下表达式根据类别或您使用 $CATEGORY 参数获取建议时指定的类别，排除物品。

```
EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)
```

以下表达式仅包含比当前物品（您在请求相关物品建议时指定的物品）便宜且由与当前物品相同的工作室创建的物品。只有当您的域用例或自定义配方生成相关项目推荐时，您才能对 CurrentItem 元素应用过滤器。

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.GENRE IN
 CurrentItem.GENRE
```

以下表达式根据多个类别字段级别排除物品。它排除 CATEGORY_L1 值为 shoe、CATEGORY_L2 值不 为 boot 的物品。

```
EXCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("shoe") AND Items.CATEGORY_L2 NOT IN
 ("boot")
```

以下表达式仅包含价格低于或等于您在使用 $PRICE 参数获取建议时指定的价格的物品。

```
INCLUDE ItemID WHERE Items.PRICE <= $PRICE
```

以下表达式仅包含早于您在获取建议时指定的时间戳（采用 Unix 纪元时间）所创建的物品。

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

以下表达式仅包含带有您在使用 $GENRE 参数获取建议时指定的一个或多个类型的物品。

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE)
```

以下表达式仅包含比当前物品更昂贵且 创建时间比您指定的时间戳（采用 Unix 纪元时间）更新的物品。如果您要获得相关物品建议，并想要根据价格和不同的创建日期应用一些特定的业务规则，则可以使用此筛选器。

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.CREATION_TIMESTAMP >
 $DATE
```

用户数据

以下表达式排除含有您在使用 $GENRE 参数获取建议时指定的一个或多个类型的物品，但前提是当前用户的年龄等于您使用 $AGE 参数获取建议时指定的值。

```
EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE) IF CurrentUser.AGE = $AGE
```

如果当前用户的年龄超过 18，则以下表达式仅包含 CATEGORY_L1 为 watch、CATEGORY_L2 为 luxury 的物品。

```
INCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("watch") AND Items.CATEGORY_L2 IN ("luxury")
 IF CurrentUser.AGE > 18
```

用户细分筛选器表达式

以下筛选条件表达式显示了如何根据物品交互数据和用户元数据筛选用户细分。它们按数据类型组织。

用户数据

以下筛选器表达式仅包含会员资格等于您在获取用户细分时指定的值的用户。

```
INCLUDE UserID WHERE Users.MEMBERSHIP_STATUS IN ($MEMBERSHIP)
```

以下筛选器表达式排除 AGE 小于您在获取用户细分时指定的值的用户。

```
EXCLUDE UserID WHERE Users.AGE < $AGE
```

物品交互数据

以下筛选器表达式仅包括单击过物品或对物品进行过评分的用户。

```
INCLUDE UserID WHERE Interactions.EVENT_TYPE IN ("click", "rating")
```

以下筛选条件表达式将这样的用户排除在用户细分外：与您在获取用户细分时指定的事件类型有物品交互。

```
EXCLUDE UserID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

操作建议筛选条件表达式示例

以下筛选条件表达式示例展示了如何根据操作交互数据、操作数据和用户数据筛选操作。它们按数据类型组织。

主题

- 操作交互数据
- 操作数据
- 用户数据

操作交互数据

以下筛选条件表达式仅在建议中包含用户已与之交互的操作，前提是这些交互具有您在获得建议时指定的事件类型。

```
INCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

以下筛选条件表达式排除了用户未根据事件类型执行的操作。

```
EXCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ("NOT_TAKEN")
```

操作数据

以下表达式根据您使用 $CATEGORY 参数获取建议时指定的一个或多个类别排除操作。

```
EXCLUDE ActionID WHERE Actions.CATEGORY IN ($CATEGORY)
```

以下表达式仅包含值大于您在获得建议时指定的值的操作。

```
INCLUDE ActionID WHERE Actions.VALUE > ($VALUE)
```

用户数据

如果当前用户拥有高级会员资格，则以下表达式仅包含高级会员的操作。

```
INCLUDE ActionID WHERE Action.MEMBERSHIP_LEVEL IN ("Premium") IF CurrentUser.MEMBERSHIP
 = $PREMIUM
```

如果当前用户是高级会员，则以下表达式排除 VALUE 小于您在获得建议时指定的值的操作。

```
EXCLUDE ActionID WHERE Actions.VALUE < ($VALUE) IF CurrentUser.MEMBERSHIP = $PREMIUM
```

组合多个表达式

要将多个表达式组合在一起，请使用竖线分隔符 (|)。如果您要使用单个筛选条件并对物品数据集和物品交互数据集或操作数据集和操作交互数据集进行筛选，请使用表达式的组合。首先对每个表达式进行独立评估，产生的结果要么是两个结果的并集，要么是交集。以下示例说明如何为物品和物品交互数据集创建表达式，但在使用操作和操作交互时，同样的规则也适用。

匹配表达式示例

如果两个表达式都使用 EXCLUDE 或两个表达式都使用 INCLUDE，则产生的结果是两个结果的并集，如下所示（A 和 B 是不同的表达式）：

- `Exclude A | Exclude B` 等于 `Exclude result from A or result from B`
- `Include A | Include B` 等于 `Include result from A or result from B`

以下示例演示了如何组合使用 INCLUDE 的两个表达式。第一个表达式仅包括含您使用 $CATEGORY 参数获取建议时指定的一个或多个类别的物品。第二个表达式包括用户已标记为 `favorite` 的物品。建议将仅包括含您指定的类别的物品以及用户已标记为收藏的物品。

```
INCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY) | INCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("favorite")
```

INCLUDE 和 EXCLUDE 示例

如果一个表达式使用 INCLUDE，另一个表达式使用 EXCLUDE，则结果是从 INCLUDE 表达式结果中减去 EXCLUDE 表达式结果，如下所示（A、B、C 和 D 是不同的表达式）。

- `Include A | Exclude B` 等于 `Include result from A - result from B`
- `Include A | Include B | Exclude C | Exclude D` 等于 `Include (A or B) - (C or D)`

表达式顺序无关紧要：如果 EXCLUDE 表达式位于 INCLUDE 表达式之前，则结果相同。

以下示例展示了如何组合 INCLUDE 表达式和 EXCLUDE 表达式。第一个表达式仅包含带有您在使用 $GENRE 参数获取建议时指定的一个或多个类型的物品。第二个表达式排除用户单击或流式传输过的物品。建议将仅包括含有您指定的类型但尚未被单击或流式传输的物品。

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE) | EXCLUDE ItemID WHERE
 Interactions.EVENT_TYPE IN ("click", "stream")
```

筛选实时建议

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包筛选实时推荐。

当您获取个性化物品建议或类似物品时，可以在请求中指定促销活动。推广 使用筛选器来定义其他业务规则，这些规则适用于可配置的建议物品子集。有关更多信息，请参阅 通过实时推荐推广物品。

主题

- 筛选实时建议（控制台）
- 筛选实时推荐 (AWS CLI)
- 筛选实时推荐 (AWS 软件开发工具包)

筛选实时建议（控制台）

要使用控制台筛选实时建议，请创建筛选器，然后将其应用于建议请求。

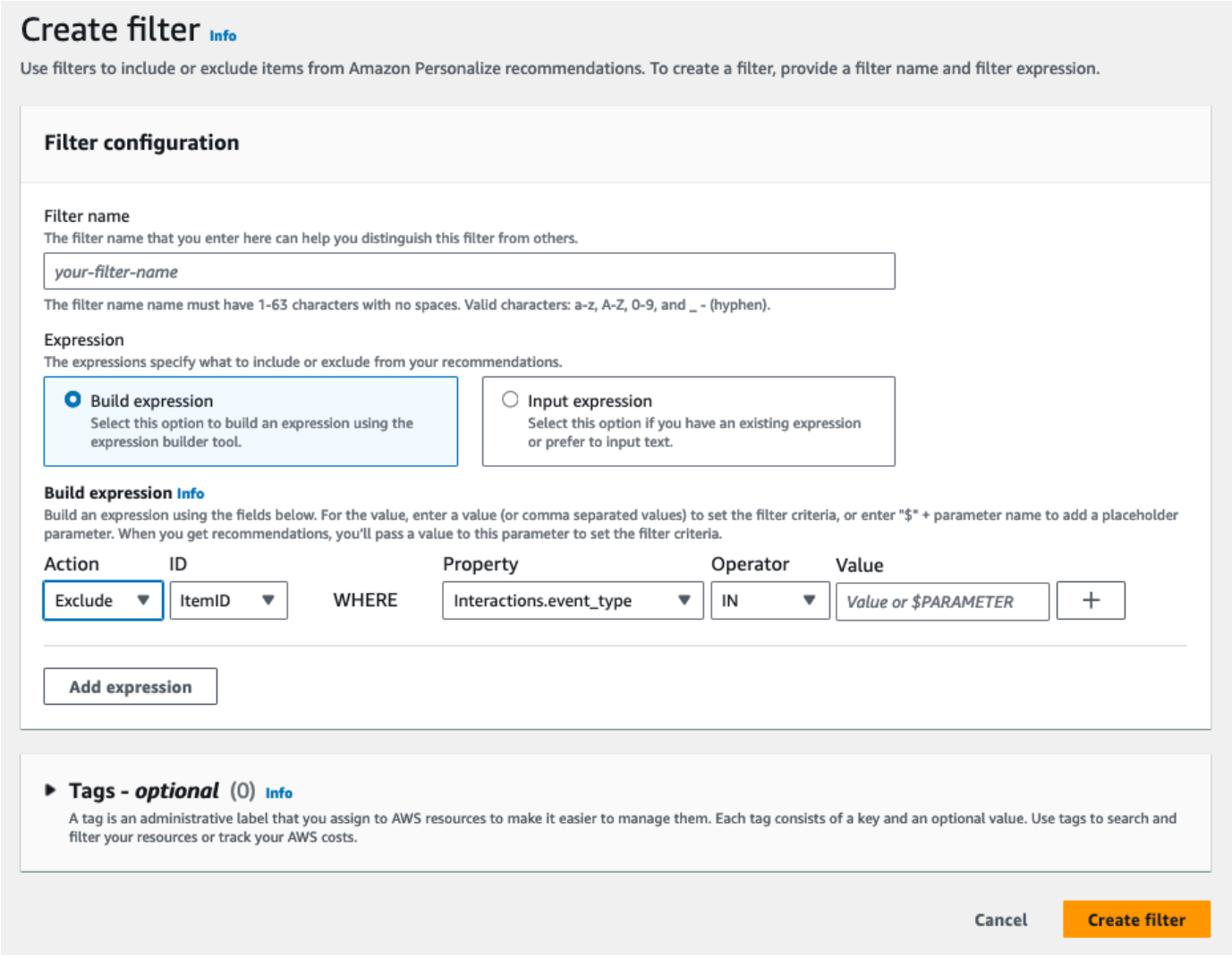
> **Note**
>
> 要使用带有参数的筛选器和您在 2020 年 11 月 10 日之前部署的市场活动来筛选建议，您必须使用 UpdateCampaign 操作重新部署市场活动或创建新市场活动。

创建筛选器（控制台）

要在控制台中创建筛选器，请选择包含要用于获取筛选建议的市场活动或推荐器的数据集组。然后，提供筛选器名称和筛选器表达式。

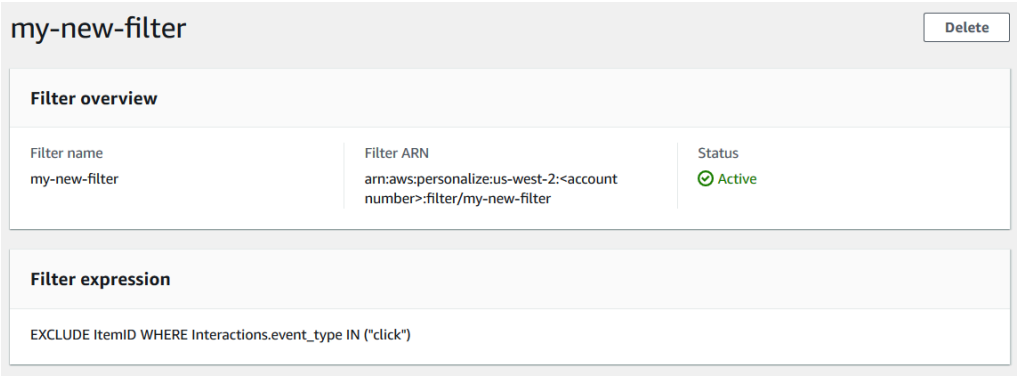
创建筛选器（控制台）

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择包含要用于获取筛选建议的市场活动或推荐器的数据集组。
3. 在导航窗格中，选择筛选条件，然后选择创建新筛选条件。这时，将显示创建筛选器页面。



4. 对于筛选器名称，输入筛选器的名称。当您将筛选器应用于建议请求时，您将按此名称选择筛选器。
5. 对于表达式，选择构建表达式或手动添加表达式，然后构建或插入表达式：
   - 要使用表达式生成器，请选择构建表达式。表达式生成器提供结构、字段和指南，用于构建格式正确的筛选器表达式。有关更多信息，请参阅 使用筛选器表达式生成器。
   - 要输入您自己的表达式，请选择手动添加表达式。有关更多信息，请参阅 筛选器表达式元素。

6. 选择结束。筛选器的概述页面显示筛选器的 Amazon 资源名称 (ARN)、状态和完整的筛选器表达式。要删除筛选器，请选择删除。有关在离开概述页面后查找和删除筛选器的信息，请参阅删除筛选器（控制台）。



应用筛选器（控制台）

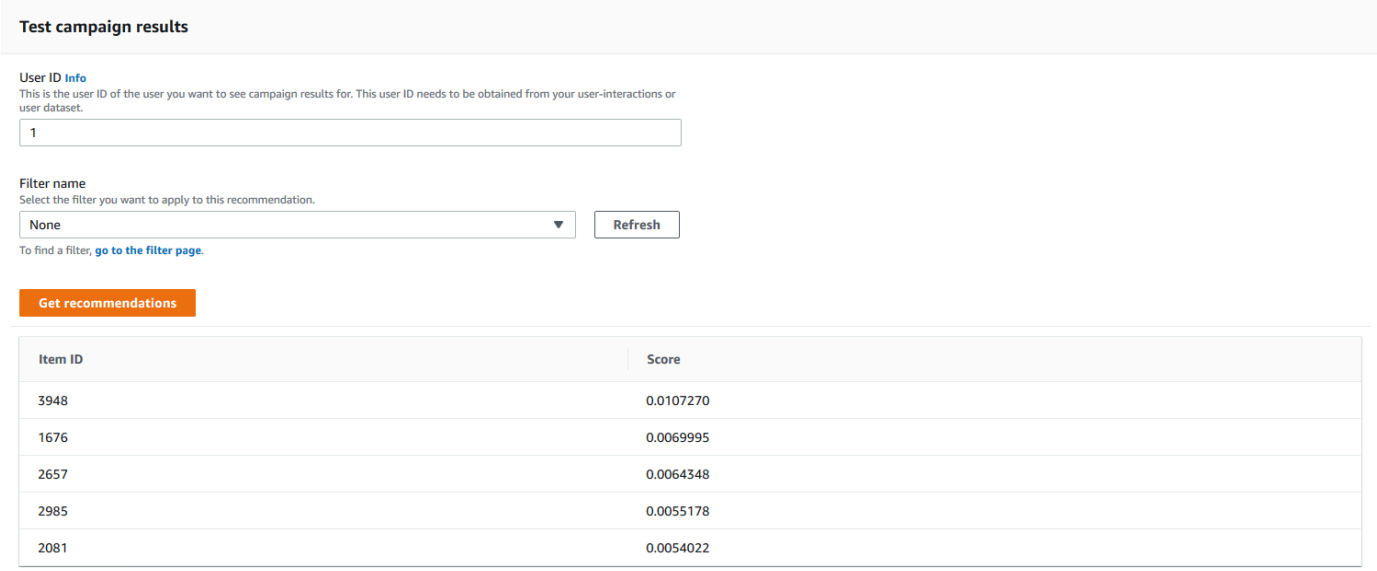
要应用筛选条件，请在测试推荐器（针对推荐器）或测试市场活动结果（针对自定义市场活动）中，选择筛选条件并输入任何筛选条件参数值。然后，获取针对用户的建议。

> **Important**
>
> 对于使用 `INCLUDE` 元素的筛选条件表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 `EXCLUDE` 元素的表达式的筛选条件，您可以省略 `filter-values`。在本例中，Amazon Personalize 未使用表达式的这一部分来筛选建议。

应用筛选器（控制台）

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择包含要用于获取筛选建议的市场活动或推荐器的数据集组。
3. 根据数据集组类型或资源类型，执行以下任一操作：
   a. 对于域数据集组，在导航窗格中选择推荐器。
   b. 对于自定义数据集组或自定义资源，在导航窗格中选择自定义资源，然后选择市场活动。
4. 在推荐器或市场活动页面上，选择目标推荐器或市场活动。

5. 为了进行比较，请先在不应用筛选条件的情况下获取建议。在测试推荐器/测试市场活动结果下，输入您要获取其建议的用户的 ID 或相关物品的物品 ID，然后选择获取建议。此时将显示一个包含前几项建议的表。

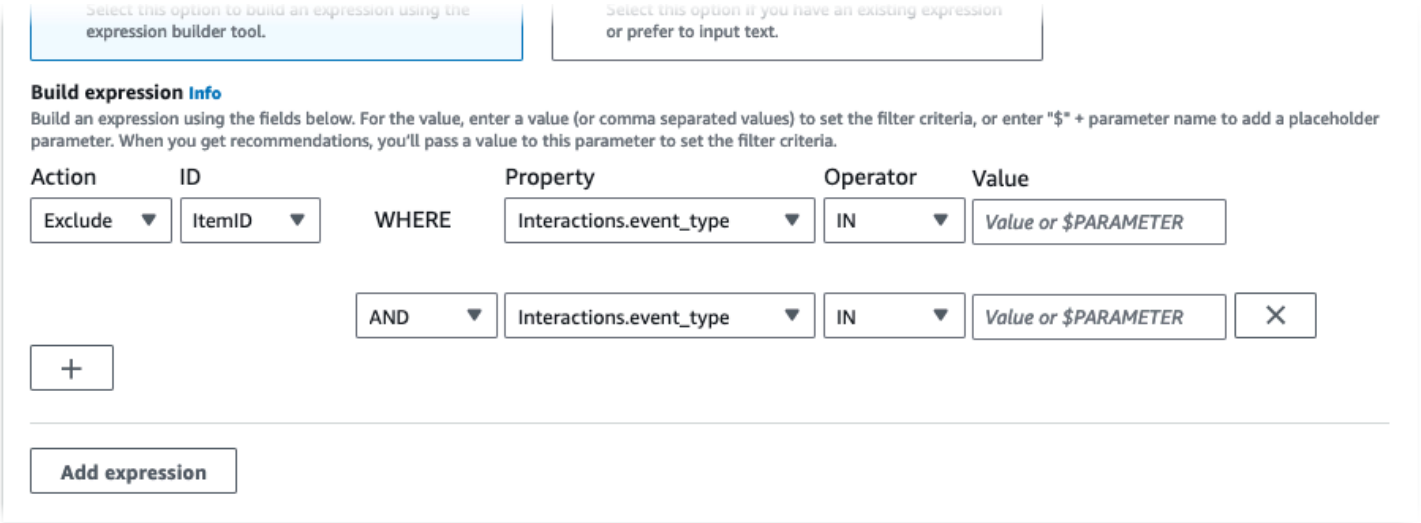


6. 从筛选器名称菜单中，选择您创建的筛选器。如果您的筛选器有任何占位符参数，则会显示每个参数的关联字段。
7. 如果您使用的筛选器带有占位符参数，请为每个参数输入值以设置筛选条件。要为一个参数使用多个值，请用英文逗号分隔每个值。
8. 使用与上一步相同的 `User ID` 或 `Item ID`，选择获取建议。此时将显示建议表。

   例如，如果用户已经购买了建议的物品，筛选条件会将其从建议列表中删除。在本示例中，物品 2657、2985 被用户未购买的最合适的物品（物品 2641 和 1573）所取代。

使用筛选器表达式生成器

创建筛选条件页面上的表达式生成器提供结构、字段和指南，用于构建格式正确的筛选条件。



构建筛选器表达式：

- 使用类型、操作、属性、运算符和值字段创建表达式。

  对于值，输入固定值，或者要在获得建议时设置筛选条件，请输入 $ + 参数名称。例如 $GENRES。获得建议时，您将提供一个或多个要筛选的值。在此示例中，将在获取建议时提供一个类型或类型列表。

  用英文逗号分隔多个非参数值。不能向筛选器中添加以逗号分隔的参数。

  > **Note**
  >
  > 选择属性（`dataset.field` 格式）后，由 AND 或 OR 条件链接的任何后续行的属性值都必须使用相同的 `dataset`。

- 使用 + 和 X 按钮，在表达式中添加或删除一行。您无法删除第一行。
- 对于新行，使用 AND 菜单上的 AND、IF 或 OR 运算符来创建条件链。

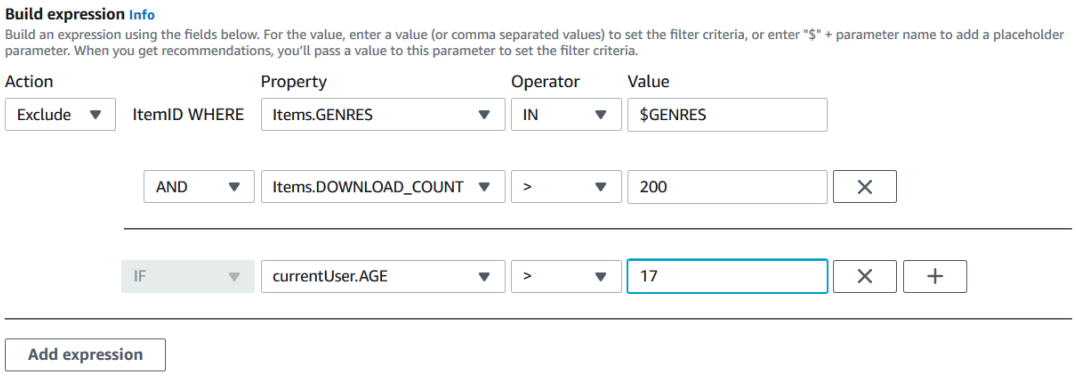
  对于 IF 条件：

  - 每个表达式只能包含一个 IF 项。如果删除 IF 条件，则表达式生成器会删除其后的所有 AND 条件。
  - 只能对按 `CurrentUser` 筛选的表达式使用 IF 条件。
- 选择添加表达式按钮，以添加其它筛选条件表达式来进行更精确的筛选。首先对每个表达式进行独立评估，产生的结果是两个结果的并集。

> **Note**
>
> 要创建同时使用物品数据集和交互数据集（或操作数据集和操作交互数据集）的筛选条件，必须使用多个表达式。

表达式生成器示例

以下示例说明如何构建筛选器，以排除含有在获得建议时指定的类型的物品（注意 $GENRES 占位符参数）。该筛选器还会排除 DOWNLOAD_COUNT 超过 200 的物品，但前提是当前用户的年龄超过 17。



删除筛选器（控制台）

删除筛选器会将筛选器从数据集组的筛选器列表中移除。

> **Important**
>
> 批量推理作业正在进行时，您无法删除筛选器。

删除筛选器（控制台）

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 从数据集组列表中，选择包含要删除的筛选器的数据集组。
3. 在导航窗格中，选择筛选器。
4. 从筛选器列表中，选择要删除的筛选器，然后选择查看详细信息。此时会显示筛选器详细信息页面。

5. 选择删除，并在确认对话框中确认删除。

筛选实时推荐 (AWS CLI)

要使用筛选推荐 AWS CLI，您可以创建一个筛选条件，然后通过在GetRecommendations或GetPersonalizedRanking请求中指定筛选条件 ARN 来应用该过滤器 ARN。

> **Important**
>
> 要使用带有参数的筛选器和您在 2020 年 11 月 10 日之前部署的市场活动来筛选建议，您必须使用 UpdateCampaign 调用重新部署市场活动或创建新市场活动。

创建过滤器 (AWS CLI)

使用以下 `create-filter` 操作创建筛选器并指定筛选器表达式。

将 `Filter name` 替换为筛选器的名称，将 `Dataset group ARN` 替换为数据集组的 Amazon 资源名称 (ARN)。将示例 `filter-expression` 替换为您自己的筛选器表达式。

```
aws personalize create-filter \
  --name Filter name \
  --dataset-group-arn dataset group arn \
  --filter-expression "EXCLUDE ItemID WHERE Items.CATEGORY IN (\"$CATEGORY\")"
```

如果成功，将显示筛选器 ARN。记下该 ARN 供将来使用。要验证筛选器是否处于活动状态，请在使用筛选器之前使用 DescribeFilter 操作。

有关 API 的更多信息，请参阅 CreateFilter。有关筛选器表达式的更多信息（包括示例），请参阅筛选条件表达式结构和元素。

应用过滤器 (AWS CLI)

使用 `get-recommendations`、`get-action-recommendations` 或 `get-personalized-ranking` 操作时，您通过将 `filter-arn` 和任何筛选条件值作为参数传递，应用筛选条件。

以下是 `get-recommendations` 操作的示例。将 `Campaign ARN` 替换为市场活动的 Amazon 资源名称 (ARN)，将 `User ID` 替换为要获得建议的用户的 ID，将 `Filter ARN` 替换为筛选器的 ARN。

如果您从推荐器（而不是市场活动）获得建议，请使用 `recommender-arn`（而不是 `--campaign-arn`），并提供推荐器的 ARN。

如果您的表达式含有任何参数，请包含 `filter-values` 对象。对于筛选器表达式中的每个参数，提供参数名称（区分大小写）和值。例如，如果您的筛选器表达式具有 $GENRE 参数，请提供 "GENRE" 作为键，并提供一个或多个类型（例如 "Comedy"）作为值。用英文逗号分隔多个值。例如 `"\"comedy\",\"drama\",\"horror\""`。

> **Important**
>
> 对于使用 `INCLUDE` 元素来包含物品的筛选器表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 `EXCLUDE` 元素排除物品的表达式的筛选器，您可以省略 `filter-values`。在本例中，Amazon Personalize 未使用表达式的这一部分来筛选建议。

```
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\""
    }'
```

删除过滤器 (AWS CLI)

要删除筛选器，请使用以下 `delete-filter` 操作。将 `filter ARN` 替换为筛选器的 ARN。

```
aws personalize delete-filter --filter-arn Filter ARN
```

筛选实时推荐 (AWS 软件开发工具包)

要使用 AWS 软件开发工具包筛选推荐，您可以创建一个过滤器，然后通过在或请求中GetRecommendations指定过滤器 ARN 来应用该过滤器 ARN。GetPersonalizedRanking

> **Important**
>
> 要使用带有参数的筛选器和您在 2020 年 11 月 10 日之前部署的市场活动来筛选建议，您必须使用 UpdateCampaign 调用重新部署市场活动或创建新市场活动。

创建过滤器 (AWS 软件开发工具包)

使用 CreateFilter 操作创建新筛选器。以下代码展示了如何创建筛选器。指定筛选器名称、数据集组的 Amazon 资源名称 (ARN)，并提供筛选器表达式。

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();
```

记录筛选器 ARN 以备后用。要验证筛选器是否处于活动状态，请在使用筛选器之前使用 DescribeFilter 操作。有关 API 的更多信息，请参阅 CreateFilter。有关筛选器表达式的更多信息（包括示例），请参阅筛选条件表达式结构和元素。

应用过滤器 (AWS 软件开发工具包)

使用 GetRecommendations、或 GetPersonalizedRanking 操作时 GetActionRecommendations，通过传递 a `filterArn` 和任意筛选值作为参数来应用过滤器。

以下代码展示了如何为用户获取经过筛选的 Amazon Personalize 物品建议。指定要获取建议的用户的 ID、市场活动的 Amazon 资源名称 (ARN)，以及筛选器的 ARN。如果您从推荐器（而不是市场活动）获得建议，请使用 `recommenderArn`（而不是 `campaignArn`），并提供推荐器的 ARN。

对于 filterValues，对于筛选器表达式中的每个可选参数，提供参数名称（区分大小写）和一个或多个值。例如，如果您的筛选器表达式具有 $GENRES 参数，请提供 "GENRES" 作为键，并提供一个或多个类型（例如 "\"Comedy"\"）作为值。对于多个值，用英文逗号分隔每个值。例如 "\"comedy\",\"drama\",\"horror\""。

> **Important**
>
> 对于使用 INCLUDE 元素来包含物品的筛选器表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 EXCLUDE 元素排除物品的表达式的筛选器，您可以省略 filter-values。在本例中，Amazon Personalize 未使用表达式的这一部分来筛选建议。

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

以下示例使用两个参数，一个参数具有两个值，另一个参数具有一个值。根据您的筛选器表达式，修改代码以添加或移除 parameterName 和 parameterValue 字段。

```
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();
```

删除过滤器 (AWS Python SDK)

要删除筛选器，请使用以下 `delete_filter` 方法。将 `filter ARN` 替换为筛选器的 ARN。

```
import boto3
personalize = boto3.client("personalize")

response = personalize.delete_filter(
  filterArn = "filter ARN"
)
```

筛选批量建议和用户细分（自定义资源）

筛选批量建议和用户细分的工作方式与筛选实时建议几乎相同。它遵循获取批量物品推荐和获取批量用户细分中描述的相同工作流程。要筛选批量建议或用户细分，请执行以下操作：

1. 创建筛选器，就像针对实时建议创建筛选器一样。有关更多信息，请参阅 筛选实时建议。
2. 准备您的输入数据并将其上传到 Amazon S3，如为批量建议准备输入数据或为用户细分准备输入数据中所述。如果您的筛选器使用占位符参数，则必须添加其他 `filterValues` 对象。有关更多信息，请参阅 在您的输入 JSON 中提供筛选器值。如果您的筛选器不使用占位符参数，则您的输入数据可以遵循批量推理作业输入和输出 JSON 示例 批量细分作业输入和输出 JSON 示例中的示例
3. 为输出数据创建一个单独的位置，可以是文件夹，也可以是其他 Amazon S3 存储桶。
4. 创建批量推理作业或批量细分作业。创建作业时，指定筛选器的 Amazon 资源名称 (ARN)。
5. 批量推理或批量细分作业完成后，从 Amazon S3 的输出位置检索建议或用户细分。

主题

- 在您的输入 JSON 中提供筛选器值
- 筛选批量工作流程（控制台）
- 筛选批处理工作流程 (AWS 软件开发工具包)

在您的输入 JSON 中提供筛选器值

对于带有占位符参数的筛选器（如 `$GENRE`），您必须在输入 JSON 的 `filterValues` 对象中提供参数的值。对于 `filterValues` 对象，每个键都是一个参数名称。每个值都是您作为参数传递的标准。用转义引号将每个值括起来：`"filterValues":{"GENRES":"\"drama\""}`。对于多个值，用英文逗号分隔每个值：`"filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}`

批量推理作业输入 JSON 示例

以下是批量推理作业 的 JSON 输入文件前几行的示例。该示例包括 `filterValues` 对象。GENRES 键对应于筛选器表达式中的 `$GENRES` 占位符。此示例中的作业使用 User-Personalization 配方。对于 RELATED_ITEMS 食谱，提供 itemId，而不是 userId。对于 PERSONALIZED_RANKING 食谱，提供 userID 和 itemList。

```
{"userId": "5","filterValues":{"GENRES":"\"horror\",\"comedy\",\"drama\""}}
{"userId": "3","filterValues":{"GENRES":"\"horror\",\"comedy\""}}
```

```
{"userId": "34","filterValues":{"GENRES":"\"drama\""}}
```

有关按食谱进行批量推理作业输入数据的更多示例，请参阅批量推理作业输入和输出 JSON 示例。您可以使用这些示例作为起点并添加上面示例中的 `filterValues` 对象。

批量细分作业输入 JSON 示例

以下是批量细分作业 的 JSON 输入文件（带筛选器值）前几行的示例。GENRES 键对应于筛选器表达式中的 $GENRES 占位符。

```
{"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\""}}
```

有关按食谱进行批量推理作业输入数据的更多示例，请参阅批量细分作业输入和输出 JSON 示例。您可以使用这些示例作为起点并添加上面示例中的 `filterValues` 对象。

筛选批量工作流程（控制台）

要使用 Amazon Personalize 控制台筛选批量工作流程，您需创建筛选器，然后创建批量推理作业或批量细分作业，并选择筛选器。有关完整的分步说明，请参阅创建批量推理作业（控制台）和创建批量细分作业（控制台）。

筛选批处理工作流程 (AWS 软件开发工具包)

要使用 AWS 软件开发工具包筛选批量推荐，请创建一个筛选条件并在CreateBatchInferenceJob或CreateBatchSegmentJob请求中包含该`FilterArn`参数。

以下代码展示了如何使用 适用于 Python (Boto3) 的 AWS SDK创建带筛选器的批量推理作业。我们建议使用不同的输出数据位置（文件夹或其他 Amazon S3 存储桶）。有关所有字段的完整说明，请参阅创建批量推理作业 (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"}}
)
```

记录实时事件以影响推荐

事件是用户与您的目录之间的交互。它可以是与物品的交互（例如用户购买物品或观看视频），也可以是执行操作（例如申请信用卡或注册会员计划）。

Amazon Personalize 可以仅根据实时事件数据、历史事件数据（或两者混合）来提供建议。记录客户与推荐交互时的实时事件。这会生成您的交互数据，并使您的数据保持最新状态。并且它会告知 Amazon Personalize 您的用户当前的兴趣，从而提高建议的相关性。

如果您的域使用案例或自定义配方支持实时个性化，则 Amazon Personalize 会实时使用事件根据用户不断变化的兴趣来更新和调整建议。

如何记录实时事件取决于您要导入的交互数据的类型：

- 对于物品交互，您可以通过 PutEvents API 操作记录实时事件。Amazon Personalize 会将此数据附加到数据集组中的物品交互数据中。有关更多信息，请参阅 记录实时物品交互事件。
- 对于操作交互，您可以使用 PutActionInteractions API 操作记录实时事件。Amazon Personalize 会将此数据附加到数据集组中的操作交互数据集。只有 PERSONALIZED_ACTIONS 配方使用操作交互数据。有关更多信息，请参阅 记录实时操作交互事件。

主题

- 实时事件如何影响建议
- 记录实时物品交互事件
- 记录实时操作交互事件
- 为匿名用户记录事件
- 第三方事件跟踪服务
- 示例实施

实时事件如何影响建议

如果您的配方支持实时个性化，则在您创建推荐器或自定义市场活动后，Amazon Personalize 会在导入后的几秒钟内为现有物品或操作使用新记录的事件数据。以下使用案例和食谱支持实时个性化：

- 为您推荐（ECOMMERCE 使用案例）
- 热门精选（VIDEO_ON_DEMAND 使用案例）

- User-Personalization-v2 配方
- User-Personalization 食谱
- Personalized-Ranking-v2 配方
- Personalized-Ranking 食谱
- Next-Best-Action 食谱

如果您使用 Trending-Now 食谱，则 Amazon Personalize 会在可配置的时间间隔内自动考虑来自新事件数据的物品。您不必创建新的解决方案版本。有关更多信息，请参阅 Trending-Now 食谱。

如果事件中的物品、操作或用户是新的物品、操作或用户，则 Amazon Personalize 如何使用数据取决于您的用户案例或配方。有关更多信息，请参阅 训练后更新数据集中的数据。

记录实时物品交互事件

物品交互事件是用户与您目录中的物品之间的交互。例如，用户购买鞋子或看电影。

在向客户展示物品建议的同时，记录实时物品交互事件。这会生成您的交互数据，并使您的数据保持最新状态。并且它会告知 Amazon Personalize 您的用户当前的兴趣，从而提高建议的相关性。

您可以使用 PutEvents API 操作记录物品交互事件。Amazon Personalize 会将事件数据附加到数据集组中的物品交互数据集。如果您记录了两个具有完全相同时间戳和相同属性的事件，则 Amazon Personalize 将只保留其中一个事件。你可以使用 AWS SDKs、 AWS Amplify 或 AWS Command Line Interface ()AWS CLI来记录物品互动事件。

如果您使用 Apache Kafka，则可以通过 Amazon Personalize Kafka 连接器将物品交互实时流式传输到 Amazon Personalize。有关信息，请参阅 personalize-kafka-connectorGithub 存储库中的 Amazon Personalize Kafka 连接器。

AWS Amplify 包括一个用于记录来自 Web 客户端应用程序的项目交互事件的 JavaScript 库和一个用于在服务器代码中记录事件的库。有关更多信息，请参阅 Amplify 文档。

主题

- 记录物品交互事件和训练模型的要求
- 创建物品交互事件跟踪器
- 记录单个物品交互事件
- 记录含事件值数据的多个物品交互事件

- 记录包含展示数据的物品交互事件
- 事件指标和归因报告

记录物品交互事件和训练模型的要求

要记录物品交互事件，您需要以下项：

- 包含一个 `Item interactions` 数据集的数据集组，该数据集可以为空。如果您浏览了入门教程指南，则可以使用您创建的同一数据集组和数据集。有关创建数据集组和数据集的信息，请参阅将训练数据导入 Amazon Personalize 数据集。
- 一个事件跟踪器。
- 调用 PutEvents API 操作。
- 如果您使用 AWS Lambda 函数调用该 PutEvents 操作，则您的函数的执行角色必须具有在`Resource`元素`*`中使用通配符执行`personalize:PutEvents`操作的权限。

您可以从空的物品交互数据集开始，记录足够的数据后，只使用新记录的事件来训练模型。对于所有使用案例（域数据集组）和食谱（自定义数据集组），您的交互数据在训练之前必须具有以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个独立用户 IDs ，每个用户至少有两次物品互动。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

创建物品交互事件跟踪器

必须先创建物品交互事件跟踪器，然后才能记录物品交互事件。事件跟踪器会将新事件数据定向到数据集组中的物品交互数据集。

使用 Amazon Personalize 控制台或 CreateEventTracker API 操作创建事件跟踪器。将以下内容作为参数传递：包含目标物品交互数据集的数据集组的 Amazon 资源名称 (ARN)。有关使用 Amazon Personalize 控制台创建事件跟踪器的说明，请参阅创建事件跟踪器（控制台）。

事件跟踪器包括跟踪 ID，您在使用PutEvents操作时将其作为参数传递。然后，Amazon Personalize 将新事件数据附加到您在事件跟踪器中指定的数据集组的物品交互数据集中。

> **Note**
>
> 您只能为一个数据集组创建一个物品交互事件跟踪器。

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'])
```

此时将显示事件跟踪器 ARN 和跟踪 ID，例如：

```
{
    "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
```

此时将显示事件跟踪器 ARN 和跟踪 ID，例如：

```
{
    "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;
}
```

记录单个物品交互事件

为数据集组创建物品交互数据集和事件跟踪器后，就可以记录物品交互事件了。以下示例显示了用于传递一个物品交互事件的 PutEvents 操作。将显示对应的架构，以及物品交互数据集中的一个示例行。

当用户首次访问您的网站或使用您的应用程序时，您的应用程序会生成唯一 sessionId。在整个会话中，您必须在所有活动中使用相同的 sessionId。Amazon Personalize 在用户登录（匿名用户）之前使用 sessionId 将事件与用户关联起来。有关更多信息，请参阅 为匿名用户记录事件。

事件列表是 Event 对象的数组。每个事件都需要一个 eventType。如果没有事件类型数据，您可以提供占位符值来满足要求。

trackingId 来自您在创建物品交互事件跟踪器中创建的事件跟踪器。userId、itemId 和 sentAt 参数映射到相应的历史 Interactions 数据集的 USER_ID、ITEM_ID 和 TIMESTAMP 字段。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

相应的数据集列

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE
Example data: user123, item-xyz, 1543631760, click
```

代码示例

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());
        }
}
```

记录含事件值数据的多个物品交互事件

为数据集组创建物品交互数据集和事件跟踪器后，就可以记录物品交互事件了。以下示例展示了如何记录具有不同事件类型和不同事件值的多个物品交互事件。

配置解决方案时，如果物品交互数据集包含 EVENT_TYPE 和 EVENT_VALUE 字段，则您可以将特定值设置为阈值，以从训练中排除记录。有关更多信息，请参阅 选择用于训练的物品交互数据。

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**
>
> 属性键使用与交互架构中的字段匹配的采用驼峰式大小写的名称。例如，如果在交互架构中定义了“NUM_RATINGS”字段，则属性键应为 numRatings。

记录包含展示数据的物品交互事件

如果您使用User-Personalization配方或将IMPRESSIONS字段添加到域数据集组中数据集的架构中，则可以在 PutEvents操作中记录展示次数数据。展示是用户与特定物品交互（例如，单击或观看）时可

见的物品列表。Amazon Personalize 使用展示次数数据来指导浏览，其中建议包括交互数据较少或相关性较低的物品。有关 Amazon Personalize 可以建模的隐式 和显式 展示的信息，请参阅展示数据。

> **Important**
>
> 如果您在 PutEvents 请求中提供了相互矛盾的隐式和显式展示数据，则 Amazon Personalize 将默认使用显式展示数据。

要记录 Amazon Personalize 建议，您需要将用户显示为展示次数数据，在 PutEvents 请求中添加 recommendationId，Amazon Personalize 会根据您的建议数据得出隐式展示次数。

要手动记录事件的展示次数数据，请在 PutEvents 命令的 impression 输入参数中列出展示。以下代码示例演示了如何使用适用于 Python 的 SDK (Boto3) 或适用于 Java 的 SDK 2.x impression 在 PutEvents 操作中包含recommendationId和。如果您同时包含这两者，则 Amazon Personalize 将默认使用显式展示。

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

使用以下 putEvents 方法记录包含展示次数数据和 recommendationId 的事件。对于 impressions 参数，将 itemID 列表作为。 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());
    }
}
```

事件指标和归因报告

要监控发送到 Amazon Personalize 的事件的类型和数量，请使用亚马逊 CloudWatch 指标。有关更多信息，请参阅 监控亚马逊 Personalize 使用亚马逊进行个性化设置 CloudWatch。

要生成显示推荐影响 CloudWatch 的报告，请创建指标归因并通过实时推荐记录用户互动。有关创建指标归因的信息，请参阅衡量 Amazon Personalize 推荐的影响。

对于每个事件，包括您向用户显示的建议的建议 ID。或者包括事件源，例如第三方。导入这些数据以比较不同的市场活动、推荐器和第三方。您最多可以导入 100 个事件归因源。

- 如果您提供 `recommendationId`，则 Amazon Personalize 会自动确定源市场活动或推荐器，并在报告的 EVENT_ATTRIBUTION_SOURCE 列中对其进行识别。
- 如果您同时提供这两个属性，则 Amazon Personalize 将仅使用 `eventAttributionSource`。
- 如果您未提供源，则 Amazon Personalize 会在报告中标记源 `SOURCE_NAME_UNDEFINED`。

以下代码显示了如何在 PutEvents 操作中`eventAttributionSource`为事件提供。

```
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)
```

以下代码显示了如何`recommendationId`为 PutEvents 操作中的事件提供。

```
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)
```

记录实时操作交互事件

操作交互事件是用户和操作之间的交互。例如，注册会员计划或申请信用卡的用户。

如果您使用 PERSONALIZED_ACTIONS 自定义配方，请记录客户与操作建议交互时的实时操作交互事件。这会生成您的交互数据，并使您的数据保持最新状态。它还会告知 Amazon Personalize 您的用户当前的兴趣，从而提高建议的相关性。只有 PERSONALIZED_ACTIONS 自定义配方使用操作交互数据。

您可以使用 PutActionInteractions API 操作记录操作交互事件。Amazon Personalize 会将此数据附加到数据集组中的操作交互数据集。

操作交互事件必须具有事件类型属性，该属性可以是以下属性之一：

- 已执行 - 当用户采取了建议的操作时，记录已执行事件。
- 未执行 – 当您的用户在查看操作后特意选择不采取该操作时，记录未执行事件。例如，当您向他们展示操作时，如果他们选择否。未执行事件可能表示客户对该操作不感兴趣。
- 已查看 - 当您在用户选择采取或不采取操作之前向他们展示操作时，记录已查看事件。Amazon Personalize 使用查看事件来了解用户的兴趣。例如，如果用户查看了某项操作但未执行该操作，则该用户将来可能对此操作不感兴趣。

您可以使用 AWS SDKs、或 AWS Command Line Interface (AWS CLI) 记录实时事件。如果您记录了两个具有完全相同时间戳和相同属性的事件，则 Amazon Personalize 将只保留其中一个事件。

主题

- 记录操作交互事件的要求
- 查找您的操作交互事件跟踪器的 ID
- 记录单个操作交互事件
- 记录多个操作交互事件

记录操作交互事件的要求

要记录实时操作交互事件，您需要以下内容：

- 包含 `Action interactions dataset` 的数据集组，该数据集可以为空。有关创建数据集组和数据集的信息，请参阅将训练数据导入 Amazon Personalize 数据集。
- 事件跟踪器的 ID。您可以在 PutActionInteractions 操作中指定此 ID。当您创建操作交互数据集时，Amazon Personalize 会自动为您创建操作交互事件跟踪器。有关更多信息，请参阅 查找您的操作交互事件跟踪器的 ID。
- 调用 PutActionInteractions 操作。

查找您的操作交互事件跟踪器的 ID

当您创建操作交互数据集时，Amazon Personalize 会自动为您创建操作交互事件跟踪器。您可以在 PutActionInteractions API 操作中指定跟踪器的 ID。Amazon Personalize 使用它来将新数据定向到数据集组中的操作交互数据集。

您可以在 Amazon Personalize 控制台的操作交互数据集的详情页面上找到事件跟踪器的 ID。你可以通过调用 DescribeDataset API 操作来找到 ID。以下 Python 代码输出操作交互数据集的跟踪 ID。

```
import boto3

personalize = boto3.client(service_name='personalize')

response = personalize.describe_dataset(
  datasetArn="Action interactions dataset ARN"
)

print(response['trackingId'])
```

记录单个操作交互事件

创建操作交互数据集后，就可以通过 PutActionInteractions 操作记录操作交互事件了。以下代码显示一个用于传递 TAKEN 事件的 `PutActionInteractions` 操作。当您向某个用户展示来自 Amazon Personalize 的建议并且他们采取操作（例如申请您的信用卡）时，您可能会记录此事件。

`actionInteractions`是一个 ActionInteraction 对象数组。`trackingId` 来自 Amazon Personalize 在您创建操作交互数据集时创建的事件跟踪器。有关更多信息，请参阅 查找您的操作交互事件跟踪器的 ID。

当用户首次访问您的网站或使用您的应用程序时，您的应用程序会生成唯一 sessionId。在整个会话中，您必须在所有活动中使用相同的 sessionId。Amazon Personalize 在用户登录（匿名用户）之前使用 sessionId 将事件与用户关联起来。有关更多信息，请参阅 为匿名用户记录事件。

userId、actionId 和 sentAt 参数映射到操作交互数据集的 USER_ID、ACTION_ID、EVENT_TYPE 和 TIMESTAMP 字段。

相应的操作交互数据集

```
USER_ID, ACTION_ID, TIMESTAMP, EVENT_TYPE
user123, action-xyz, 1543631760, TAKEN
```

代码示例

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'
  }]
)
```

记录多个操作交互事件

以下代码显示如何为具有相同 sessionId 的同一个用户记录多个操作交互事件。

相应的操作交互数据集

```
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'
  }]
)
```

为匿名用户记录事件

> **Important**
>
> 如果您没有为用户记录至少一个含 sessionId 和 userId 的事件，则 Amazon Personalize 不会在训练时使用仅按 sessionId 跟踪的活动。而且，训练完成后，将不再根据按 sessionId 跟踪的活动提出建议。这将为 userId 在登录前后创建连续的事件历史记录。

在用户创建账户之前，您可以记录他们的物品交互或操作交互事件。为匿名用户记录事件，以构建持续的事件历史记录，其中包含此类用户登录前后的事件。这为 Amazon Personalize 提供了更多关于用户的交互数据，这有助于生成更多相关的建议。

要记录匿名用户（尚未登录的用户）的事件，只需为每个事件指定一个 sessionId。当用户首次访问您的网站或使用您的应用程序时，您的应用程序会生成唯一 sessionId。在整个会话中，您必须在所有活动中使用相同的 sessionId。Amazon Personalize 在用户登录之前使用 sessionId 将事件与用户关联起来。

Amazon Personalize 在训练时不会使用来自匿名用户的事件，除非您将他们与 userId 关联起来。有关更多信息，请参阅 为匿名用户构建持续的事件历史记录。

要为匿名用户提供实时个性化设置，请在您的或请求中将 sessionID 指定为用户 ID。GetRecommendations GetActionRecommendations

- 有关显示如何记录与 PutEvents 操作以及会话 ID 和用户 ID 的项目交互事件的代码示例，请参阅。记录单个物品交互事件
- 有关显示如何使用 PutActionInteractions 操作以及 sessionID 和用户 ID 记录动作交互事件的代码示例，请参阅。记录单个操作交互事件

为匿名用户构建持续的事件历史记录

要为匿名用户构建事件历史记录并让 Amazon Personalize 在训练时使用他们的事件，请至少记录一个同时含 `sessionId` 和 `userId` 的事件。然后，您可以为 `userId` 记录任意数量的事件。在开始提供 `userId` 之后，`sessionId` 可以改变。在下一次全面的再训练中，Amazon Personalize 会将 `userId` 与按原始 `sessionId` 跟踪的匿名用户历史记录关联起来。

再训练完成后，将根据按匿名事件的 `sessionId` 跟踪的活动以及按 `userId` 跟踪的任何事件，提出建议。

> **Note**
>
> 如果您的用户没有创建账户，而您希望 Amazon Personalize 在训练时使用这些数据，则您可以使用 `sessionId` 作为事件中的 `userId`。但是，如果用户最终创建了一个账户，则您将无法将匿名浏览中的事件与他们的新 `userId` 关联起来。

第三方事件跟踪服务

以下客户数据平台 (CDPs) 可以帮助您从应用程序中收集事件数据并将其发送到 Amazon Personalize。

- Amplitude - 您可以使用 Amplitude 来跟踪用户操作，以帮助您了解用户的行为。有关使用 Amplitude 和 Amazon Personalize 的信息，请参阅以下 AWS 合作伙伴网络 (APN) 博客文章：使用 Amplitude 和 Amazon Personalize 衡量个性化的有效性。
- Segment - 您可以使用 Segment 将您的数据发送到 Amazon Personalize。有关将 Segment 与 Amazon Personalize 集成的更多信息，请参阅 Amazon Personalize 目标。

示例实施

有关展示如何使用 Amazon Personalize 对使用事件跟踪器和PutEvents操作的用户实时行为做出反应的 Jupyter 笔记本示例，请参阅存储库 getting_started 文件夹中的 2.view_campaign_and_ interactions.ipynb。amazon-personalize-samples GitHub

有关展示如何直播用户与推荐互动的事件的示例，请参阅 Amazon Personalize 示例存储库中的 streaming_e vents。 GitHub

有关包含要在您的 Amazon Personalize 资源和客户端应用程序之间进行实时 APIs 部署的源代码和支持文件的完整示例，请参阅 AWS 示例 GitHub 存储库 APIs中的实时个性化。该项目包括如何实施以下内容：

- 用户上下文和用户事件收集
- 响应缓存
- 根据物品元数据装饰建议
- A/B 测试
- API 身份验证

维护建议相关性

随着目录的增长，相关建议可以提高应用程序的用户参与度、点击率和转化率。要维护和提高针对用户的 Amazon Personalize 建议相关性，请更新数据和自定义资源。这样，Amazon Personalize 可以从用户的最新行为中学习，并在建议中加入您的最新物品。

主题

- 使数据集保持最新状态
- 维护域推荐器
- 维护自定义解决方案

使数据集保持最新状态

随着目录的增长，使用批量或单个数据导入操作来更新历史数据。有关导入历史数据的更多信息，请参阅将训练数据导入 Amazon Personalize 数据集。有关训练模型后导入的数据如何影响建议的信息，请参阅训练后更新数据集中的数据。

对于可提供个性化实时建议的使用案例和配方，将您的物品交互数据集与用户的行为保持同步。为此，可以使用事件跟踪器和 PutEvents API 操作记录物品交互。Amazon Personalize 会根据您的用户在与您的目录交互时的最新活动来更新建议。有关实时个性化的信息，请参阅实时个性化。有关记录实时事件的更多信息，请参阅记录实时事件以影响推荐。

维护域推荐器

Amazon Personalize 每 7 天自动重新训练一次支持推荐器的模型。这是一次全面的再训练，基于数据集中的全部数据创建全新的模型。如果您修改训练中使用的列，则 Amazon Personalize 会自动开始对支持推荐器的模型进行全面的重新训练。

- 对于热门精选 和为您推荐 使用案例，Amazon Personalize 会更新您的推荐器，以考虑推荐新物品。自动更新并不是模型从用户行为中学习的完整再训练。相反，自动更新允许 Amazon Personalize 在推荐器进行下一次全面再训练之前在建议中加入您的新物品。有关自动更新的信息，请参阅自动更新。
- 如果您使用当前趋势 使用案例，则 Amazon Personalize 会每两小时自动评估一次您的交互数据，并识别热门物品。您不必等待推荐器重新训练。

推荐器重新训练过程中，您仍可从推荐器中获得建议。在重新训练完成之前，推荐器使用之前的配置和模型。要跟踪更新，您可以在 Amazon Personalize 控制台的推荐器详细信息页面上，查看最新推荐器更新的时间戳。或者，您可以查看 DescribeRecommender 操作的 `latestRecommenderUpdate` 详细信息。

维护自定义解决方案

默认情况下，所有新解决方案都使用自动训练，每 7 天创建一个新的解决方案版本。训练会持续运行，直到您删除解决方案。

创建解决方案时，我们建议使用自动训练来管理解决方案版本创建。自动训练使维护解决方案变得更加容易。有了自动训练，解决方案从较新数据中学习就不再需要手动干预。如果没有自动训练，则必须手动为解决方案创建新的解决方案版本，这样才能从最新数据中学习。有关配置自动训练的更多信息，请参阅配置自动训练。

训练频率取决于业务需求、所使用的配方以及导入数据的频率。对于所有配方，我们建议至少每周进行一次训练。对于自动训练，这是默认的训练频率。如果您经常添加新物品或操作，则可能需要更高的训练频率，具体取决于您的配方。

- 如果您使用 User-Personalization-v2、User-Personalization 或 Next-Best-Action，则解决方案会自动更新为考虑在推荐中加入新物品或操作。自动更新与自动训练不同。自动更新不会创建全新的解决方案版本，模型也不会从最新数据中学习。为了维持您的解决方案，训练频率仍然应该是至少每周一次。有关自动更新的更多信息（包括其他准则和要求），请参阅自动更新。
- 如果您使用 Trending-Now，则 Amazon Personalize 会在可配置的时间间隔内自动识别交互数据中最热门的物品。Trending-Now 可以通过批量交互数据或流式处理交互数据推荐自上次训练以来添加的物品。您的训练频率仍然应该是至少每周一次。有关更多信息，请参阅 Trending-Now 食谱。
- 如果您不使用具有自动更新的配方或 Trending-Now 配方，则 Amazon Personalize 仅在下一次训练之后考虑在推荐中加入新物品。例如，如果您使用 Similar-Items 配方并每天添加新物品，则必须使用每日训练频率，这些物品才会出现在当天的推荐中。

训练后更新数据集中的数据

随着目录的增长，将额外训练数据导入数据集。这样有助于维持和提高 Amazon Personalize 推荐的相关性。您可以使用批量或单独数据导入操作导入更多数据。

- 使用单独导入时，Amazon Personalize 会将新记录附加到数据集中。要更新单个物品、用户或操作，可以导入具有相同 ID 但属性经过修改的记录。每次导入操作最多可以导入 10 条记录。

  有关单独导入记录的更多信息，请参阅将单个记录导入 Amazon Personalize 数据集。有关记录实时事件的信息，请参阅记录实时事件以影响推荐。
- 使用批量导入时，您可以通过创建另一个导入任务来添加或替换批量数据。默认情况下，数据集导入作业会替换您批量导入的数据集中的任何现有数据。相反，您可以通过更改作业的导入模式，将新记录附加到现有数据中。

  要使用数据集导入作业将数据附加到物品交互数据集或操作交互数据集，您必须至少有 1000 条新的物品交互或操作交互记录。在完成批量导入后 20 分钟内，Amazon Personalize 会使用您的新批量数据，更新您在数据集组中创建的所有筛选条件。此更新允许 Amazon Personalize 在筛选针对用户的建议时使用最新数据。

创建物品或用户数据集后，您可以将其架构替换为新的或现有的架构。如果数据结构在您创建数据集后发生了更改，则您可以替换该数据集的架构。例如，您可能会有想让 Amazon Personalize 在训练期间考虑的新物品元数据列。或者，您可能想要添加一列仅在筛选建议时使用的数据。有关更多信息，请参阅 替换数据集的架构，以便添加新列。

创建推荐器或自定义解决方案版本后，新数据如何影响推荐取决于数据的类型、导入方法以及所用的域应用场景或自定义配方。以下各节说明了新数据如何影响下一次训练之前的实时和批量推荐。

主题

- 新数据如何影响实时建议
- 新数据如何影响批量推荐（自定义资源）

新数据如何影响实时建议

创建推荐器或自定义解决方案版本后，新数据如何影响实时推荐取决于数据的类型、导入方法以及所用的域应用场景或自定义配方。以下各节说明了新数据如何影响下一次训练之前的实时推荐。

训练可以是推荐器的每周自动训练，也可以是自动或手动解决方案版本创建。对于使用 User-Personalization 的手动训练，请省略 `trainingMode` 以使用默认 `FULL` 训练模式。

主题

- 新的交互
- 新物品
- 新用户
- 新操作

新的交互

新交互是您在最近的训练后导入的物品或操作交互。对于实时和批量数据，如果交互涉及新物品或操作，则在配方或应用场景具备浏览功能时 Amazon Personalize 可能会考虑未经训练就在推荐中加入新物品或新操作。有关更多信息，请参阅新物品或新操作。

实时事件

对于具有实时个性化特征的应用场景和配方，Amazon Personalize 会立即使用最新训练中存在的用户和物品或操作之间的实时交互。在事件中为用户生成推荐时，Amazon Personalize 会使用这些实时交互。有关实时个性化的更多信息，请参阅实时个性化。

对于任何不具有实时个性化特征的域应用场景和自定义配方（例如推荐类似物品），您的模型仅在训练后从实时交互数据中学习。

批量交互

对于批量交互，针对增量和完整数据集导入作业，您的模型仅在下一次训练之后，才从批量物品交互或操作交互数据中学习。批量数据不用于更新实时个性化建议。

有关导入更多批量数据的更多信息，请参阅通过数据集导入作业将批量数据导入到 Amazon Personalize。

新物品

新物品是您在最近一次训练后导入的物品。可以来自交互数据或物品数据集中的物品元数据。

考虑在建议中加入新物品，如下所示：

- 对于热门精选 和为您推荐 域场景或 User-Personalization-v2、User-Personalization 或 Next-Best-Action 配方，Amazon Personalize 每两小时自动更新一次模型。每次更新后，Amazon Personalize

都会在浏览过程中考虑推荐新物品。在考虑新物品时，Amazon Personalize 会考虑该物品的所有元数据。但是，只有在您记录了物品的交互并训练新模型之后，这些数据才会对推荐产生更大的影响。有关更新的信息，请参阅自动更新。

- 如果您使用当前趋势 使用案例，则 Amazon Personalize 会每两小时自动评估一次您的交互数据，并识别热门物品。您不必等待推荐器进行训练。如果您使用 Trending-Now 配方，则 Amazon Personalize 会在可配置的时间间隔内未经训练就自动考虑所有新物品。有关配置时间间隔的信息，请参阅 Trending-Now 食谱。
- 如果您不使用 Trending-Now 配方，或者您的应用场景或配方不支持自动更新，则 Amazon Personalize 将只在下一次训练之后考虑新物品。

新用户

新用户是指您在最近一次训练后导入的用户。可以来自交互数据或用户数据集中的用户元数据。对于新的匿名用户（没有 userId 的用户），您可以通过 `sessionId` 为用户记录事件，Amazon Personalize 将在用户登录之前将事件与用户关联。有关更多信息，请参阅 为匿名用户记录事件。

Amazon Personalize 会按如下方式为新用户生成建议：

- 如果您使用当前当前趋势域使用案例或 Trending-Now 自定义食谱，则新用户会立即收到热门物品的建议。如果您使用 Popularity-Count 食谱，则新用户会立即收到交互次数最多的物品的建议。
- 对于为用户提供个性化建议的食谱或使用案例，针对新用户的建议基于现有用户的早期交互历史记录。系统更有可能将这些现有用户与之交互的第一批物品或操作推荐给新用户。对于 User-Personalization 或 Personalized-Ranking 食谱，如果您将 `recency_mask` 设置为 `true`，则建议还包括基于交互数据中最新流行趋势的物品。

以下内容可以增加建议对新用户的相关性：

- 交互数据 - 提高新用户建议相关性的主要方法是从他们与物品的交互中导入数据。有关新交互数据如何影响建议的信息，请参阅新的交互。
- 用户元数据 - 导入用户元数据（例如 GENDER 或 MEMBERSHIP_STATUS）可以改善建议。要使元数据影响建议，您必须等待域推荐器的每周自动再训练完成。或者，您必须手动创建新的解决方案版本。
- 上下文元数据 - 如果您的使用案例或配方支持上下文元数据，并且您的物品交互数据集包含上下文数据的元数据字段，则您可以在请求建议时提供用户的上下文。这不需要再训练。有关更多信息，请参阅 利用上下文元数据提高建议的相关性。

新操作

新操作是自上次训练以来您导入的操作。它们可以来自操作交互数据，也可以来自操作数据集中的操作。

借助 Next-Best-Action 配方，Amazon Personalize 每两小时自动更新一次解决方案版本。每次更新后，Amazon Personalize 都会在浏览过程中考虑推荐新操作。在考虑新操作时，Amazon Personalize 会考虑该操作的所有元数据。但是，只有在您记录了操作的操作交互并进行了完全再训练之后，这些数据才会对建议产生更大的影响。有关更新的信息，请参阅自动更新

新数据如何影响批量推荐（自定义资源）

创建自定义解决方案版本后，新数据如何影响批量推荐取决于数据的类型、导入方法和所用的自定义配方。

对于用户细分，Amazon Personalize 仅使用上次完整解决方案版本训练中提供的数据生成细分。而且，Amazon Personalize 仅使用您以 FULL（替换现有数据）导入模式导入的批量数据。有关用户细分的更多信息，请参阅使用自定义资源获取批量用户细分。

生成批量物品推荐时，Amazon Personalize 会考虑在创建最新解决方案版本时存在的所有批量数据。可以使用“完整”或“增量”导入模式，导入这些数据。要使较新的批量记录影响批量建议，您必须创建新的解决方案版本，然后创建批量推理作业。

下面几节介绍单个导入如何影响批量物品推荐。

主题

- 新的交互
- 新用户
- 新物品

新的交互

如果您使用 USER_PERSONALIZATION 或 PERSONALIZED_RANKING 配方，则 Amazon Personalize 会在数据导入后约 15 分钟内考虑现有物品和用户的新物品交互数据。在最新训练中必须存在这些物品和用户。为确保考虑事件，我们建议您至少等待 15 分钟，然后再开始批量推理作业。对于所有其他配方，以及对于具有新物品或用户的事件，您必须为流事件创建新的解决方案版本，以便影响批量推荐。

新用户

对于没有交互数据的用户，推荐最初仅针对热门物品。如果您使用 USER_PERSONALIZATION 或 PERSONALIZED_RANKING 配方，并为用户记录事件，则其推荐可能会在导入后约 15 分钟内变得更加相关，而无需重新训练。为确保考虑事件，我们建议您至少等待 15 分钟，然后再开始批量推理作业。对于所有其他食谱，您必须为流事件创建新的解决方案版本，以影响用户的批量建议，而无需交互数据。

新物品

使用 User-Personalization-v2 和 User-Personalization 时，当您创建批量推理作业并为解决方案指定经过全面训练的最新解决方案版本时，Amazon Personalize 会自动更新解决方案版本，以便在浏览推荐中加入新物品。如果您不指定最新解决方案版本，则不会进行更新。对于任何其他食谱，您都必须创建一个新的解决方案版本，以便在批量建议中包含新物品。有关浏览的更多信息，请参阅探索。

替换数据集的架构，以便添加新列

创建物品或用户数据集后，您可以将其架构替换为新的或现有的架构。如果数据结构在您创建数据集后发生了更改，则您可以替换该数据集的架构。例如，您可能会有想让 Amazon Personalize 在训练期间考虑的新物品元数据列。或者，您可能想要添加一列仅在筛选建议时使用的数据。

替换数据集的架构时，必须保留先前架构中的所有字段，并且不能更改其数据类型或属性。在您替换数据集的架构后，Amazon Personalize 会自动将任何新列排除在任何现有推荐器或自定义解决方案的训练之外。有关更多指南和要求，请参阅指南和要求。

您可以用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 和 AWS 软件开发工具包替换数据集的架构。

主题

- 指南和要求
- 替换数据集的架构（控制台）
- 替换数据集的架构 (AWS CLI)
- 替换数据集的架构 (AWS 软件开发工具包)

指南和要求

在替换数据集架构之前，确保您了解以下指南和要求：

- 您无法替换物品交互数据集、操作交互数据集或操作数据集的架构。
- 您可以向替换架构中添加新字段，但必须保留先前架构中的所有字段。而且，您无法更改其数据类型或属性。例如，如果之前的架构包含分类字符串数据的 MEMBERSHIP_STATUS 字段，则您使用的新架构必须包含具有这些属性和数据类型的 MEMBERSHIP_STATUS 字段。
- 如果当前架构有要重命名的字段，或者您要更改其数据类型或属性，则您可以添加一个具有新名称和修改类型或属性的新字段。然后，在训练中加入新字段，排除旧字段。任何新字段都必须支持 null 数据。如果旧字段不支持空数据，则在导入数据时，您可以使用占位符数据来确保您的导入与架构相匹配。有关配置推荐器使用的列的信息，请参阅更新推荐器。有关配置解决方案使用的列的信息，请参阅配置训练时使用的列。
- 任何新字段都必须支持 null 数据。有关向字段添加空类型的信息，请参阅架构数据类型。
- 在您替换数据集的架构后，Amazon Personalize 会自动将任何新列排除在任何现有推荐器或自定义解决方案的训练之外。使用修改后的数据集涉及以下操作：

- 要在训练中使用任何新列，请导入与新架构一致的数据。然后，更新所有推荐器以使用任何新列，或者创建新的自定义解决方案并配置训练时使用的列。

  有关更新推荐器使用的列的信息，请参阅更新推荐器。有关配置解决方案使用的列的信息，请参阅配置训练时使用的列。
- 要仅在筛选时使用任何列，请导入与新架构一致的数据，创建使用新数据的筛选器，然后将筛选器应用于建议请求。您无需更新任何推荐器，也无需创建或更新任何自定义资源。

替换数据集的架构（控制台）

要通过 Amazon Personalize 控制台替换数据集的架构，您需选择要修改的数据集，然后选择替换为新架构或使用现有架构。

替换数据集的架构

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 在数据集组页面，选择您的数据集组。
3. 在导航窗格中，选择数据集，然后选择要修改的数据集对应的单选按钮。
4. 选择操作，然后选择替换架构。
5. 在架构详细信息中，选择替换为新架构或以前创建的架构。
6. 指定要使用的新架构。如果您选择了：
   - 替换为新架构，接着为架构命名，并在架构定义中对架构 JSON 进行更改。
   - 使用先前创建的架构，然后对于先前创建的架构，选择要使用的架构。这时，仅会列出符合条件的架构。有关架构要求的信息，请参阅指南和要求。
7. 选择替换。当数据集处于活动状态时，您可以开始导入与新架构一致的数据。有关更多信息，请参阅 将训练数据导入 Amazon Personalize 数据集。

替换数据集的架构 (AWS CLI)

要将数据集的架构替换为 AWS CLI，请使用`update-dataset`命令，指定要更新的数据集的 Amazon 资源名称 (ARN) 和要使用的新架构的 ARN。您无法更新物品交互数据集、操作交互数据集或操作数据集的架构。

以下代码显示了如何使用 AWS CLI更新数据集的架构。要将数据集的架构替换为新架构，请先使用 `create-schema` 命令。然后，使用以下代码将当前架构替换为新架构。有关使用创建架构的信息 AWS CLI，请参阅创建数据集和架构 (AWS CLI)。有关数据集和架构要求的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。

```
aws personalize update-dataset \
--dataset-arn Dataset ARN \
--schema-arn New schema ARN
```

当数据集处于活动状态时，您可以开始导入与新架构一致的数据。有关更多信息，请参阅 将训练数据导入 Amazon Personalize 数据集。有关数据集最新更新的信息，您可以使用 DescribeDataset 操作。

替换数据集的架构 (AWS 软件开发工具包)

要将数据集的架构替换为 AWS 软件开发工具包，请使用 `UpdateDataset` API 操作。指定要更新的数据集的 Amazon 资源名称 (ARN) 和要使用的新架构。您无法更新物品交互数据集、操作交互数据集或操作数据集的架构。

以下代码显示如何将数据集的架构替换为 适用于 Python (Boto3) 的 AWS SDK。要将数据集的架构替换为新架构，请先使用 CreateSchema 操作。然后，使用以下代码将当前架构替换为新架构。有关使用 AWS 软件开发工具包创建架构的信息，请参阅创建数据集和架构 (AWS SDKs)。有关数据集和架构要求的信息，请参阅为 Amazon Personalize 架构创建架构 JSON 文件。

```
import boto3

personalize = boto3.client('personalize')

update_dataset_response = personalize.update_dataset(
    datasetArn = 'dataset_arn',
    schemaArn = 'new_schema_arn'
)

print(update_dataset_response)
```

当数据集处于活动状态时，您可以开始导入与新架构一致的数据。有关更多信息，请参阅 将训练数据导入 Amazon Personalize 数据集。有关数据集最新更新的信息，您可以使用 DescribeDataset 操作。

将数据集中的训练数据导出到 Amazon S3

将数据导入 Amazon Personalize 数据集后，您可以将数据导出到 Amazon S3 存储桶。您可以导出数据，来验证和检查 Amazon Personalize 用来生成建议、实时查看您之前记录的物品交互事件或对数据进行离线分析的数据。

您可以选择仅导出批量导入的数据（使用 Amazon Personalize 数据集导入任务导入），也可以选择仅导出您单独导入的数据（使用控制台或者 `PutEvents`、`PutUsers` 或 `PutItems` 操作导入的记录），或者两者兼而有之。

> **Note**
>
> 您无法导出操作交互数据集或操作数据集中的数据。

对于与所有字段 完全匹配的记录，Amazon Personalize 仅导出一条记录。如果两条记录具有相同的 ID，但一个或多个字段不同，则 Amazon Personalize 会根据您选择导出的数据包含或删除记录：

- 如果您同时导出批量数据和增量数据，则 Amazon Personalize 将仅导出具有相同 ID 的最新物品（在物品数据集导出中），并仅导出具有相同 ID 的用户（在用户数据集导出中）。对于物品交互数据集，Amazon Personalize 会导出所有物品交互数据。
- 如果您仅导出增量数据，则 Amazon Personalize 会导出您单独导入的所有物品、用户或物品交互数据，包括具有相同 ID 的物品或用户。只有与所有字段完全匹配的记录才会被排除在外。
- 如果您仅导出批量数据，则 Amazon Personalize 会包括您批量导入的所有物品、用户或物品交互数据，包括具有相同 ID 的物品或用户。只有与所有字段完全匹配的记录才会被排除在外。

要导出数据集，请创建数据集导出作业。数据集导出作业 是一种记录导出工具，可将数据集中的记录输出到 Amazon S3 存储桶中的一个或多个 CSV 文件。输出 CSV 文件包括标题行，其列名称与数据集架构中的字段相匹配。

主题

- 数据集导出作业权限要求
- 在 Amazon Personalize 中创建数据集导出作业

数据集导出作业权限要求

要导出数据集，Amazon Personalize 需要将文件添加到 Amazon S3 存储桶的权限。要授予权限，请将新的 AWS Identity and Access Management (IAM) 策略附加到您的 Amazon Personalize 服务角色，该策略授予该角色对您的存储桶使用 `PutObject` 和 `ListBucket` 操作的权限，并将存储桶策略附加到您的输出 Amazon S3 存储桶，该策略授予 Amazon Personalize 主体使用 `PutObject` 和 `ListBucket` 操作的权限。

如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

用于导出数据集的服务角色策略

以下示例策略授予您的 Amazon Personalize 服务角色使用 `PutObject` 和 `ListBucket` 操作的权限。将 `amzn-s3-demo-bucket` 替换为输出存储桶的名称。有关将策略附加到 IAM 服务角色的信息，请参阅将 Amazon S3 策略附加到您的 Amazon Personalize 服务角色中。

```
{
    "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/*"
            ]
        }
    ]
}
```

用于导出数据集的 Amazon S3 存储桶策略

以下示例策略授予 Amazon Personalize 针对 Amazon S3 存储桶使用 `PutObject` 和 `ListBucket` 操作的权限。将 `amzn-s3-demo-bucket` 替换为您的存储桶的名称。有关将 Amazon S3 存储桶策略添

加到存储桶的信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用 Amazon S3 控制台添加存储桶策略。

```
{
    "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/*"
            ]
        }
    ]
}
```

在 Amazon Personalize 中创建数据集导出作业

您可以使用 Amazon Personalize 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包创建数据集导出任务。

创建数据集导出作业（控制台）

将数据导入数据集并创建输出 Amazon S3 存储桶后，您可以将数据导出到存储桶进行分析。要使用 Amazon Personalize 控制台导出数据集，您需要创建数据集导出作业。有关创建 Amazon S3 存储桶的信息，请参阅《Amazon Simple Storage Service 用户指南》中的创建存储桶。

在导出数据集之前，确保您的 Amazon Personalize 服务角色可以访问和写入您的输出 Amazon S3 存储桶。请参阅数据集导出作业权限要求。

创建数据集导出作业（控制台）

1. 打开 Amazon Persalize 控制台，网址为https://console.aws.amazon.com/personalize/home。

2. 在导航窗格中，选择数据集组。
3. 在数据集组页面，选择您的数据集组。
4. 在导航窗格中，选择数据集。
5. 选择要导出到 Amazon S3 存储桶的数据集。
6. 在数据集导出作业中，选择创建数据集导出作业。
7. 在数据集导出作业详细信息中，对于数据集导出作业名称，输入导出作业的名称。
8. 对于 IAM 服务角色，选择您在为 Amazon Personalize 创建 IAM 角色中创建的 Amazon Personalize 服务角色。
9. 对于 Amazon S3 数据输出路径，输入目标 Amazon S3 存储桶。使用以下语法：

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

10. 如果您使用 AWS KMS 加密，请在 KMS 密钥 ARN 中输入密钥的亚马逊资源名称 (ARN)。 AWS KMS
11. 对于导出数据类型，根据您最初导入数据的方式，选择要导出的数据类型。

    - 选择批量，仅导出使用数据集导入作业批量导入的数据。
    - 选择增量，仅导出您使用控制台或者 `PutEvents`、`PutUsers` 或 `PutItems` 操作单独导入的数据。
    - 选择两者，导出数据集中的所有数据。

12. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
13. 选择创建数据集导出作业。

    在数据集概述页面的数据集导出作业中，作业以导出作业状态列出。当状态为活动时，表示数据集导出作业完成。然后，您可以从输出 Amazon S3 存储桶中下载数据。有关从 Amazon S3 存储桶下载对象的信息，请参阅《Amazon Simple Storage Service 用户指南》中的下载对象。

创建数据集导出任务 (AWS CLI)

将数据导入数据集并创建输出 Amazon S3 存储桶后，您可以将数据集导出到存储桶进行分析。要使用导出数据集 AWS CLI，请使用`create-dataset-export-job` AWS CLI 命令创建数据集导出任务。有关创建 Amazon S3 存储桶的信息，请参阅《Amazon Simple Storage Service 用户指南》中的创建存储桶。

在导出数据集之前，确保 Amazon Personalize 服务角色可以访问和写入您的输出 Amazon S3 存储桶。请参阅数据集导出作业权限要求。

以下是 create-dataset-export-job AWS CLI 命令的示例。为作业命名，将 dataset arn 替换为要导出的数据集的 Amazon 资源名称 (ARN)，然后将 role ARN 替换为您在为 Amazon Personalize 创建 IAM 角色中创建的 Amazon Personalize 服务角色的 ARN。在中 s3DataDestinationkmsKeyArn，可以选择为您的 AWS KMS 密钥提供 ARN，为path提供输出 Amazon S3 存储桶的路径。

对于 ingestion-mode，通过以下选项指定要导出的数据：

- 指定 BULK，仅导出使用数据集导入作业批量导入的数据。
- 指定仅导PUT出您使用控制台或PutEvents PutUsers、或PutItems操作单独导入的数据。
- 指定 ALL，导出数据集中的所有数据。

有关更多信息，请参阅 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
```

这时，将显示数据集导出作业 ARN。

```
{
  "datasetExportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-export-job/
DatasetExportJobName"
}
```

使用 DescribeDatasetExportJob 操作检查状态。

```
aws personalize describe-dataset-export-job \
  --dataset-export-job-arn dataset export job ARN
```

创建数据集导出任务 (AWS 软件开发工具包)

将数据导入数据集并创建输出 Amazon S3 存储桶后，您可以将数据集导出到存储桶进行分析。要使用 AWS 软件开发工具包导出数据集，请使用CreateDatasetExportJob操作创建数据集导出任务。有关创建 Amazon S3 存储桶的信息，请参阅《Amazon Simple Storage Service 用户指南》中的创建存储桶。

以下代码展示了如何使用 适用于 Python (Boto3) 的 AWS SDK 或 SDK for Java 2.x SDK 创建数据集导出任务。

在导出数据集之前，确保 Amazon Personalize 服务角色可以访问和写入您的输出 Amazon S3 存储桶。请参阅数据集导出作业权限要求。

SDK for Python (Boto3)

使用以下 `create_dataset_export_job`，将数据集中的数据导出到 Amazon S3 存储桶。为作业命名，将 `dataset arn` 替换为要导出的数据集的 Amazon 资源名称 (ARN)，然后将 `role` ARN 替换为您在为 Amazon Personalize 创建 IAM 角色中创建的 Amazon Personalize 服务角色的 ARN。在中 `s3DataDestinationkmsKeyArn`，可以选择为您的 AWS KMS 密钥提供 ARN，为`path`提供输出 Amazon S3 存储桶的路径。

对于 `ingestionMode`，通过以下选项指定要导出的数据：

- 指定 `BULK`，仅导出使用数据集导入作业批量导入的数据。
- 指定仅导`PUT`出您使用控制台或`PutEvents` PutUsers、或`PutItems`操作单独导入的数据。
- 指定 `ALL`，导出数据集中的所有数据。

```
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

使用以下 `createDatasetExportJob` 方法创建数据集导出作业。将以下内容作为参数传递：a PersonalizeClient、导出任务的名称、要导出的数据集的 ARN、提取模式、输出 Amazon S3 存储桶的路径以及密钥的 ARN。 AWS KMS

`ingestionMode` 可以是以下一个选项：

- 使用 `IngestionMode.BULK`，仅导出使用数据集导入作业批量导入的数据。
- 用于`IngestionMode.PUT`仅导出您使用控制台或`PutEvents` PutUsers、或`PutItems`操作单独导入的数据。
- 使用 `IngestionMode.ALL`，导出数据集中的所有数据。

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

删除 Amazon Personalize 资源的要求

删除资源有助于避免不必要的费用。例如，当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关费用和价格的完整列表，请参阅 Amazon Personalize 定价。

要使用 Amazon Personalize 控制台删除资源，请在资源的详细信息页面上选择删除。要使用 Amazon Personalize API 删除资源，可以将 `Delete` API 与 SDK 或 AWS Command Line Interface（AWS CLI）一起使用。

有关使用 Amazon Personalize 控制台、AWS Command Line Interface（AWS CLI）或 AWS SDK 删除数据集的详细步骤，请参阅删除数据集以删除其所有数据。您可以将这些步骤中的模式应用于其他 Amazon Personalize 资源。有关从数据集组中删除用户及其数据的信息，请参阅使用数据删除作业删除用户及其数据。

必须先删除一些资源，然后才能删除其他资源。例如，如果您创建了事件跟踪器和物品交互数据集，则必须先删除事件跟踪器，然后才能删除数据集。以下几个部分介绍了删除 Amazon Personalize 资源的准则和顺序要求。

主题

- 删除资源的准则
- 资源删除的推荐顺序
- 使用数据删除作业删除用户及其数据
- 删除数据集以删除其所有数据

删除资源的准则

以下是删除资源的准则：

- 在 Amazon Personalize 中删除资源是一项不可逆的操作。删除开始后就无法停止。
- 无法删除正从一种状态变为另一种状态的资源。例如，您无法删除其状态为 CREATE PENDING 或 IN PROGRESS 的资源。资源状态必须为 ACTIVE 或 CREATE FAILED。这包括解决方案的 `latestSolutionUpdate` 状态。可以使用 `Describe` API 检查资源的状态。例如，使用 DescribeCampaign API 操作。
- 有关删除 Amazon S3 中的训练数据的信息，请参阅如何从 S3 存储桶删除对象？。

- 数据集导入作业完成后，就不需要继续为此支付费用，并且您无法删除它们。
- 您无需为架构付费，也无法使用 Amazon Personalize 控制台删除架构。要删除架构，请使用 DeleteSchema API 操作。

以下是删除数据集的特定要求：

- 在删除任何数据集之前，必须先删除所有筛选器。
- 如果创建了事件跟踪器，则在删除物品交互数据集之前，必须删除事件跟踪器。
- 如果创建了引用该数据集的指标归因，则必须先删除指标归因。
- 如果您使用 User-Personalization-v2、User-Personalization 或 Next-Best-Action 配方或热门精选和为您推荐应用场景，则删除数据集会暂停任何关联的解决方案版本或推荐器的自动更新。
- 无关联的 `DatasetImportJob` 可以处于 CREATE PENDING 或 IN PROGRESS 状态。
- 无关联的 `BatchInferenceJob` 或 `BatchSegmentJob` 可以处于 CREATE PENDING 或 IN PROGRESS 状态。
- 关联的 `Recommender` 和 `SolutionVersion` 不能处于 CREATE PENDING 或 IN PROGRESS 状态。
- 关联的 `Campaign` 不能处于 CREATE PENDING 或 IN PROGRESS 状态。

资源删除的推荐顺序

为避免删除错误，我们建议您按以下顺序从数据集组中删除资源。要识别数据集组中的资源，您可以使用 List API 操作。例如，您可以使用 ListFilters API 操作来识别数据集组中的所有筛选器。

1. 任何市场活动或推荐器 - 要使用 API 删除市场活动或推荐器，请使用 DeleteCampaign 或 DeleteRecommender API 操作。使用推荐器时，您可以停止推荐器并稍后再启动。这样，您就可以暂停推荐器计费，并仅在使用推荐器时才付费。有关更多信息，请参阅 停止推荐器。
2. 任何解决方案 - 要使用 API 删除解决方案，请使用 DeleteSolution API 操作。要删除一个解决方案，则不能有正在进行中的解决方案更新。其 `latestSolutionUpdate` 状态必须为 ACTIVE 或 CREATE FAILED。删除解决方案会删除所有关联的解决方案版本。所有解决方案版本都不能处于 CREATE PENDING 或 IN PROGRESS 状态。
3. 事件跟踪器 - 要使用 API 删除事件跟踪器，请使用 DeleteEventTracker API 操作。必须先删除事件跟踪器，然后才可以删除物品交互数据集。
4. 指标归因 - 要使用 API 删除指标归因，请使用 DeleteMetricAttribution API 操作。

5. 所有筛选器 - 要使用 API 删除筛选器，请使用 DeleteFilter API 操作。必须先删除所有筛选器，然后才能删除数据集。
6. 任何数据集 - 要使用 API 删除数据集，请使用 DeleteDataset API 操作。
7. 数据集组 - 要使用 API 删除数据集组，请使用 DeleteDatasetGroup API 操作。
8. 架构 - 要删除架构，请使用 DeleteSchema API 操作。

使用数据删除作业删除用户及其数据

导入数据后，您可以从数据集组中删除用户及其数据，包括元数据和交互数据。为了遵循合规计划，或者为了应对用户删除请求，或者为了在用户群变化时使数据保持最新，您可能需要删除用户数据。

删除用户后，Amazon Personalize 将不再根据他们的数据进行训练，在生成用户细分时也不再考虑这些用户。

要删除对 Amazon Personalize 数据集中的用户以及数据集组中的模型的引用，请执行以下操作：

1. 准备一个 CSV 文件，在 USER_ID 列中列出要删除的用户的用户 ID。
2. 将 CSV 文件上传到 Amazon S3 存储桶。您的 Amazon Personalize 服务角色必须具有访问此存储桶的权限。
3. 创建数据删除作业。数据删除作业是一个批处理作业，用于从数据集组中的模型和数据集中删除用户及其数据。

主题

- 指南和要求
- 准备待删除用户的列表
- 创建数据删除作业

指南和要求

以下是删除用户的准则和要求：

- 在创建数据删除作业之前，请确保没有正在使用您的数据集的作业，例如训练作业、批处理作业，或批量或单个导入操作。并避免在执行数据删除作业时创建此类作业。如果进行了任何训练或导入，我们无法保证用户的数据会从模型中删除，因此我们建议再创建一个数据删除作业。

- 数据删除作业不会删除对 Amazon Personalize 以外的用户的引用。例如，该作业不会从 Amazon S3 存储桶中的批量推荐中删除他们的用户 ID。您必须手动删除这些记录。
- 对于一个状态为 PENDING 的数据集组，您最多可以有 5 个删除作业。
- 一个或多个数据删除输入文件的最大总大小为 100 MB。在创建删除作业时，您可以重复使用相同的输入文件。
- 每个数据删除作业都会删除数据集组中的用户及其交互数据。要删除他们在所有数据集组中的数据，必须为每个数据集组创建一个数据删除任务。
- 创建作业后，最多可能需要一天时间才能从数据集和模型中删除用户的数据。
- 作业完成后，请务必更新所有自定义资源。请务必创建新的解决方案版本，并在必要时更新市场活动。如果您使用自动训练，您仍然可以手动创建新的解决方案版本。
- 您的 Amazon Personalize 服务角色必须有权访问包含待删除用户列表的 Amazon S3 存储桶。它需要存储桶及其内容的 `GetObject` 和 `ListBucket` 权限。这些权限与导入数据时的权限相同。有关授予权限和策略示例的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。
- 您不能在存储待删除用户的用户 ID 列表的 Amazon S3 存储桶上使用自己的 AWS Key Management Service 密钥。
- 如果某个物品仅在您的物品交互数据集中出现，并且只有您要删除的用户与该物品进行了互动，则该物品将不再出现在推荐中。

准备待删除用户的列表

在从 Amazon Personalize 删除用户之前，必须在 CSV 文件中准备好待删除用户的列表并将文件上传到 Amazon S3。

准备待删除用户的列表并上传

1. 创建一个 CSV 文件，列出待删除用户的用户 ID。下面显示了 CSV 文件必须如何格式化。

```
USER_ID
abc
2a
5basc
ab35
123f
a55d
0v22
441fa
```

```
efg
```

2. 将 CSV 文件上传到 Amazon Simple Storage Service（Amazon S3）存储桶。有关将文件上传到 Amazon S3 的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用拖放功能上传文件和文件夹。
3. 向 Amazon Personalize 授予访问存储桶和 CSV 文件的权限。Amazon Personalize 必须有权对您的存储桶及其内容执行 `GetObject` 和 `ListBucket` 操作。这些权限与导入数据时的权限相同。有关授予权限和策略示例的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

创建数据删除作业

完成准备待删除用户的列表后，就可以通过数据删除作业来删除用户。

数据删除作业是一个批处理作业，用于从数据集组中的模型和数据集中删除用户及其数据。删除用户后，Amazon Personalize 将不再根据他们的数据进行训练，在生成用户细分时也不再考虑这些用户。

创建数据删除作业时，需要指定待删除用户的列表在 Amazon S3 中的位置。

- 如果您的数据位于单个文件中，请为 Amazon S3 位置使用以下语法：

  `s3://amzn-s3-demo-bucket/<folder path>/<CSV filename>.csv`
- 如果您的 CSV 文件位于 Amazon S3 存储桶的某个文件夹中，则可以指定该文件夹的路径。对于数据删除作业，Amazon Personalize 会使用该文件夹及任何子文件夹中带有 `.csv` 文件扩展名的所有文件。该作业会忽略任何其他类型的文件。使用以下语法，在文件夹名称后面加上 `/`：

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

您使用的角色必须有权对您的 Amazon S3 存储桶及其内容执行 `GetObject` 和 `ListBucket` 操作。有关授予权限和策略示例的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

您可以使用 Amazon Personalize 控制台、AWS Command Line Interface（AWS CLI）或 AWS SDK 创建数据删除作业。

创建数据删除作业（控制台）

要使用 Amazon Personalize 控制台删除用户，请创建一个包含名称、IAM 服务角色和 Amazon S3 数据位置的数据删除作业。

删除记录（控制台）

1. 打开 Amazon Personalize 控制台（网址为 https://console.aws.amazon.com/personalize/home），并登录您的账户。
2. 在数据集组页面，选择您的数据集组。这时，将显示数据集组概述。
3. 在导航窗格中，选择数据集。
4. 在数据删除作业中，选择创建作业。
5. 在作业详细信息中，为作业取名。
6. 在 S3 输入源中，对于 S3 位置，指定用于存储待删除用户的用户 ID 列表的 CSV 文件在 Amazon S3 中的位置。在准备待删除用户的列表中准备这个文件。
7. 对于 IAM 角色，选择创建新角色或使用现有角色。如果您完成了为 Amazon Personalize 创建角色的先决条件，并授予此角色访问 Amazon S3 存储桶的权限，请选择使用现有服务角色，并指定您在为 Amazon Personalize 创建 IAM 角色中创建的角色。

   您使用的角色必须有权对您的 Amazon S3 存储桶及其内容执行 `GetObject` 和 `ListBucket` 操作。这些权限与导入数据时的权限相同。有关授予权限和策略示例的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。
8. 对于标签，可以选择添加任何标签。有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。
9. 请选择创建任务。作业开始，并且会显示详细信息页面。

   创建作业后，需要大约一天时间才能从数据集和模型中删除用户的数据。在作业完成之前，Amazon Personalize 会继续在训练时使用这些数据。用户可能会出现在用户细分中。

   当状态显示为 COMPLETED 时，数据删除完成。如果作业因任何原因而失败，我们建议创建另一个数据删除作业。作业完成后，请务必更新所有自定义资源。请务必创建新的解决方案版本，并在必要时更新市场活动。如果您使用自动训练，您仍然可以手动创建新的解决方案版本。

创建数据删除作业（AWS CLI）

要使用 AWS CLI 删除用户，请使用 `create-data-deletion-job` 命令。此命令使用 CreateDataDeletion API 操作。以下代码展示了如何创建数据删除作业。要使用该代码，请更新它以指定作业名称、您在为 Amazon Personalize 创建 IAM 角色中创建的 IAM 角色以及 Amazon S3 数据位置。在准备待删除用户的列表中准备这个文件。

```
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
```

创建作业后，需要大约一天时间才能从数据集和模型中删除用户的数据。在作业完成之前，Amazon Personalize 会继续在训练时使用这些数据。用户可能会出现在用户细分中。

当状态为 COMPLETED 时，作业完成。使用 `describe-data-deletion-job` 命令检查状态，并指定数据删除作业 ARN。有关该 API 操作的更多信息，请参阅 DescribeDataDeletionJob。要查看按创建时间排序的数据删除作业的历史记录，请使用 ListDataDeletionJobs API 操作。

如果作业因任何原因而失败，我们建议创建另一个数据删除作业。作业完成后，请务必更新所有自定义资源。请务必创建新的解决方案版本，并在必要时更新市场活动。如果您使用自动训练，您仍然可以手动创建新的解决方案版本。

创建数据删除作业（AWS SDK）

要使用 AWS SDK 删除用户，请使用 CreateDataDeletionJob API 操作。以下代码展示了如何创建数据删除作业。要使用该代码，请更新它以指定作业名称、您在为 Amazon Personalize 创建 IAM 角色中创建的 IAM 角色以及 Amazon S3 数据位置。在准备待删除用户的列表中准备这个文件。

```
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'])
```

创建作业后，需要大约一天时间才能从数据集和模型中删除用户的数据。在作业完成之前，Amazon Personalize 会继续在训练时使用这些数据。用户可能会出现在用户细分中。

当状态为 COMPLETED 时，作业完成。使用 DescribeDataDeletionJob 操作检查状态，并指定数据删除作业 ARN。要查看按创建时间排序的数据删除作业的历史记录，请使用 ListDataDeletionJobs API 操作。

如果作业因任何原因而失败，我们建议创建另一个数据删除作业。作业完成后，请务必更新所有自定义资源。请务必创建新的解决方案版本，并在必要时更新市场活动。如果您使用自动训练，您仍然可以手动创建新的解决方案版本。

删除数据集以删除其所有数据

要删除数据集中的所有数据，可以删除数据集。您可以使用 Amazon Personalize 控制台删除数据集，AWS Command Line Interface (AWS CLI) 或 AWS SDKs。在删除数据集之前，请注意以下事项：

- 在删除任何数据集之前，必须先删除所有筛选器。
- 如果创建了事件跟踪器，则在删除物品交互数据集之前，必须删除事件跟踪器。
- 如果创建了引用该数据集的指标归因，则必须先删除指标归因。
- 如果您使用 User-Personalization-v 2、用户个性化设置、 Next-Best-Action食谱或首选用例以及推荐给您的用例，则删除数据集会暂停任何关联的解决方案版本或推荐者的自动更新。
- 无关联的 `DatasetImportJob` 可以处于 CREATE PENDING 或 IN PROGRESS 状态。
- 无关联的 `BatchInferenceJob` 或 `BatchSegmentJob` 可以处于 CREATE PENDING 或 IN PROGRESS 状态。
- 关联的 `Recommender` 和 `SolutionVersion` 不能处于 CREATE PENDING 或 IN PROGRESS 状态。
- 关联的 `Campaign` 不能处于 CREATE PENDING 或 IN PROGRESS 状态。

主题

- 删除数据集（控制台）
- 删除数据集 (AWS CLI)
- 删除数据集 (AWS SDKs)

删除数据集（控制台）

要使用 Amazon Personalize 控制台删除数据集，请导航至数据集详情页面并选择删除。

删除数据集

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台。
2. 在导航窗格中，选择数据集组。
3. 在数据集组页面，选择您的数据集组。
4. 在导航窗格中，选择数据集。
5. 选择数据集以打开其详细信息页面。
6. 在数据集的详细信息页面上，选择删除并确认数据集删除。

删除数据集 (AWS CLI)

以下代码演示如何使用 AWS CLI 和DeleteDataset操作删除数据集。

```
aws personalize delete-dataset --dataset-arn dataset-arn
```

删除数据集 (AWS SDKs)

以下代码演示如何使用 AWS SDKs 和DeleteDataset操作删除数据集。

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());
    }
}
```

衡量 Amazon Personalize 推荐的影响

当您的客户与推荐交互时，您可以衡量 Amazon Personalize 推荐如何帮助您实现目标。您可以确定哪些市场活动和推荐器对关键绩效指标影响更大。例如，您可以确定哪个资源产生的观看时间最长、点击数最多或购买数最多。您还可以将 Amazon Personalize 建议的效果与第三方服务生成的建议效果进行比较。

当您知道哪个市场活动或推荐器产生的影响更大时，您可以采取行动来进一步从其推荐中受益。例如，您可以提高用户对网站上的推荐内容的兴趣，以提高参与度。或者，您也可以在市场活动中展示这些推荐，例如个性化电子邮件或定向广告。

如果您发现某个资源没有产生预期的影响，则可以采取措施来改进推荐。例如，您可以使用 Amazon Personalize 控制台来分析用于创建资源的训练数据，以便改进数据，然后再次导入数据。有关分析数据的更多信息，请参阅分析 Amazon Personalize 数据集中的数据质量和数量。

以下内容可帮助您衡量建议的影响：

- 指标归因：Amazon Personalize 指标归因可根据您指定的指标以及您导入的物品交互和物品数据创建报告。例如，用户观看的影片总时长或单击事件的总数。在您创建指标归因后，Amazon Personalize 会自动向亚马逊发送有关来自 PutEvents API 操作的事件的指标和增量批量数据 CloudWatch。对于批量数据，您可以选择将报告发布到 Amazon S3 存储桶。
- A/B 测试：使用 Amazon Personalize 推荐进行 A/B 测试涉及向不同的用户群展示不同类型的推荐并比较结果。您可以使用 A/B 测试来帮助比较和评估不同的推荐策略、评估模型性能以及衡量建议的影响。

主题

- 使用指标归因衡量建议的影响
- 通过 A/B 测试衡量推荐的影响

使用指标归因衡量建议的影响

要衡量物品建议的影响，您可以创建指标归因。指标归因可根据您导入的物品交互和物品数据以及您指定的指标创建报告。例如，用户观看的影片总时长或单击事件的总数。Amazon Personalize 在 15 分钟的时间段内汇总计算结果。对于流媒体互动数据和增量批量数据，Amazon Personalize 会自动向亚马逊发送指标报告 CloudWatch。对于批量数据，您可以选择将报告发布到 Amazon S3 存储桶。

对于您导入的每个交互，都要包含源数据，以比较不同的市场活动、推荐器和第三方。您可以包括向用户或事件源（例如第三方）展示的建议的建议 ID。

例如，您可能有一个视频流应用程序，它显示来自两个不同 Amazon Personalize 推荐器的电影推荐。如果您想查看哪个推荐器产生的观看事件最多，可以创建指标归因来跟踪观看事件的总数。然后，您可以在用户与建议交互时记录观看事件，并将 `recommendationId` 包含在每个事件中。Amazon Personalize 使用 `recommendationId` 来识别每个推荐器。在记录事件时，您可以查看中两个推荐者在每 15 分钟内汇总的观看活动总数。 CloudWatch有关显示如何为事件添加 `recommendationId` 或 `eventAttributionSource` 的代码示例，请参阅事件指标和归因报告。

主题

- 指标归因的准则和要求
- 创建 Amazon Personalize 指标归因
- 更新 Amazon Personalize 指标归因
- 删除 Amazon Personalize 指标归因
- 在中查看指标数据的图表 CloudWatch
- 将指标归因报告发布到 Amazon S3

指标归因的准则和要求

只有在您创建指标归因后，Amazon Personalize 才会开始计算和报告建议的影响。要构建最完整的历史记录，我们建议您在导入交互数据之前创建指标归因。当您使用 Amazon Personalize 控制台为物品交互数据集创建数据集导入作业时，您可以选择在新选项卡中创建指标归因。然后，您可以返回导入作业，将其完成。

在创建指标归因并记录事件或导入增量批量数据后，每个指标每月将产生一些 CloudWatch 费用。有关 CloudWatch 定价的信息，请参阅 Amazon CloudWatch 定价页面。要停止向发送指标 CloudWatch，请删除指标归因。

要了解建议在一段时间内的影响，请在客户与建议交互时继续导入数据。如果您已经导入了数据，仍然可以创建指标归因并开始衡量建议的影响。但是，Amazon Personalize 不会报告您在创建数据之前导入的数据。

以下是生成带有指标归因的报告的指南和要求：

- 您必须授予 Amazon Personalize 访问和输入数据的权限 CloudWatch。有关策略示例，请参阅 让 Amazon Personalize 访问权限 CloudWatch。

- 要将指标发布到 Amazon S3，请向 Amazon Personalize 授予写入存储桶的权限。您还必须在指标归因中提供存储桶路径。有关策略示例，请参阅 向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限。
- 要向发布指标 CloudWatch，记录必须少于 14 天。如果您的数据较旧，则这些记录将不会包含在计算或报告中。
- 导入重复的事件（与所有属性完全匹配的事件）可能会导致意外行为，包括不准确的指标。我们建议您在导入之前从任何批量数据中删除重复的记录，并避免通过 `PutEvents` 操作导入重复的事件。
- 您的物品交互数据集必须包含 `EVENT_TYPE` 列。
- 您无法为操作交互数据集中的数据创建指标报告。
- 每个数据集组最多可以创建一个指标归因。每个指标归因最多可以有 10 个指标。

要比较源，每个交互事件都必须包含 `recommendationId` 或 `eventAttributionSource`。您最多可以提供 100 个唯一的事件归因源。有关 `PutEvents` 代码示例，请参阅 事件指标和归因报告。

- 如果您提供 `recommendationId`，则 Amazon Personalize 会自动确定源市场活动或推荐器，并在报告的 EVENT_ATTRIBUTION_SOURCE 列中对其进行识别。
- 如果您同时提供这两个属性，则 Amazon Personalize 将仅使用 `eventAttributionSource`。
- 如果您未提供源，则 Amazon Personalize 会在报告中标记源 `SOURCE_NAME_UNDEFINED`。

主题

- 让 Amazon Personalize 访问权限 CloudWatch
- 向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限

让 Amazon Personalize 访问权限 CloudWatch

> **Important**
>
> 当您授予权限时，Amazon Personalize 会在其中放置并验证少量数据。 CloudWatch这将产生不到 0.30 美元的一次性费用。有关 CloudWatch 定价的更多信息，请参阅 Amazon CloudWatch 定价页面。

要向 Amazon Personalize 提供访问权限 CloudWatch，请将新 AWS Identity and Access Management (IAM) 政策附加到您的 Amazon Personalize 服务角色，授予该角色使用PutMetricData操作的权限 CloudWatch。以下策略示例授予 PutMetricData 权限。

JSON

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

向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限

向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限：

- 将 IAM 策略附加到您的 Amazon Personalize 服务角色，以向该角色授予针对存储桶使用 PutObject 操作的权限。

  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/*"
```

```
            ]
        }
    ]
}
```

- 将存储桶策略附加到您的输出 Amazon S3 存储桶，以向 Amazon Personalize 主体授予使用 `PutObject` 操作的权限。

  如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 [授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限]。

  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/*"
            ]
        }
    ]
}
```

创建 Amazon Personalize 指标归因

> **Important**
>
> 在创建指标归因并记录事件或导入增量批量数据后，每个指标每月将产生一些 CloudWatch 费用。有关 CloudWatch 定价的信息，请参阅 Amazon CloudWatch 定价页面。要停止向发送指标 CloudWatch，请删除指标归因。

要开始生成指标报告，您需要创建指标归因并导入交互数据。在创建指标归因时，您需指定要报告的事件类型列表。对于每种事件类型，您需指定一个 Amazon Personalize 在收集数据时应用的函数。可用函数包括 `SUM(DatasetType.COLUMN_NAME)` 和 `SAMPLECOUNT()`。

例如，您可能有一个在线视频流应用程序，并想要跟踪以下两个指标：建议的点击率和观看的电影总时长，其中，物品数据集中的每个视频都包含一个 `LENGTH` 属性。您将创建一个指标归因并添加两个指标，其中每个指标都有事件类型和函数。第一个可能是带有 `SAMPLECOUNT()` 函数的 `Click` 事件类型。第二个可能是带有 `SUM(Items.LENGTH)` 函数的 `Watch` 事件类型。

您只能将 `SUM()` 函数应用于物品数据集和物品交互数据集的数值列。要将 `SUM()` 函数应用于物品数据集中的列，您必须先导入物品元数据。

您可以使用 Amazon Personalize 控制台或 AWS SDK 创建指标归因。 AWS Command Line Interface

主题

- 创建指标归因（控制台）
- 创建指标归因 (AWS CLI)
- 创建指标归因 (AWS 软件开发工具包)

创建指标归因（控制台）

要使用 Amazon Personalize 控制台创建指标归因，请导航到指标归因页面，然后选择创建指标归因。创建指标归因时，您需指定可选的 Amazon S3 存储桶路径、Amazon Personalize IAM 服务角色以及要报告的指标列表。

当您使用 Amazon Personalize 控制台创建物品交互数据集导入作业时，您可以选择在新选项卡中创建指标归因。然后，您可以返回导入作业，将其完成。如果您已进入配置指标归因页面，则可以跳至步骤 4。

创建指标归因

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择您的数据集组。
3. 在导航窗格的自定义资源下，选择指标归因。
4. 在指标归因详细信息中，选择创建指标归因。
5. 在配置指标归因页面上，为指标归因命名。
6. 如果您想将指标发布到 Amazon S3，则对于 Amazon S3 数据输出路径，输入目标 Amazon S3 存储桶。这样，您就可以选择在每次创建数据集导入作业时发布指标。使用以下语法：

   s3://amzn-s3-demo-bucket/<folder> path>
7. 如果您使用 AWS KMS 加密，请在 KMS 密钥 ARN 中输入密钥的亚马逊资源名称 (ARN)。 AWS KMS 您必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。
8. 在 IAM 角色中，选择创建新服务角色或使用现有角色。您选择的角色必须具有PutMetricData权限 CloudWatch。如果您想发布到 Amazon S3，则该角色必须拥有访问 Amazon S3 存储桶的 PutObject 权限。

   要使用您在中创建的角色为 Amazon Personalize 创建 IAM 角色，您可能需要为 CloudWatch 和 Amazon S3 添加策略。

   有关策略示例，请参阅让 Amazon Personalize 访问权限 CloudWatch和向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限。
9. 选择下一步。
10. 在定义指标属性页面上，选择如何定义指标。选择构建指标属性以使用生成器工具。选择输入指标属性，以 JSON 格式输入指标。

    - 如果您选择构建指标属性，请为每个指标提供名称、事件类型并选择一个函数。对于 SUM() 函数，选择列名称。选择添加指标属性以添加其他指标。
    - 如果您选择输入指标属性，请以 JSON 格式输入每个指标。下面演示了如何设置指标格式。

```
{
    "EventType": "watch",
    "MetricName": "MinutesWatchedTracker",
    "MetricMathExpression": "SUM(Items.LENGTH)"
```

```
}
```

11. 选择下一步。
12. 在查看并创建页面上，查看新指标归因的详细信息。要进行更改，请选择 Previous。要创建指标归因，请选择创建。指标属性为活动后，您可以开始导入数据并查看结果。有关查看结果的信息，请参阅在中查看指标数据的图表 CloudWatch。有关将结果发布到 Amazon S3 的信息，请参阅将指标归因报告发布到 Amazon S3。

创建指标归因 (AWS CLI)

以下代码演示如何使用 AWS Command Line Interface创建指标归因。您指定的角色必须拥有PutMetricData访问您的 Amazon S3 存储桶的 CloudWatch 权限，如果发布到 Amazon S3，则必须拥有访问您的 Amazon S3 存储桶的PutObject权限。要使用您在中创建的角色为 Amazon Personalize 创建 IAM 角色，您可能需要为 CloudWatch 和 Amazon S3 添加策略。有关策略示例，请参阅让 Amazon Personalize 访问权限 CloudWatch和向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限。

为每个指标指定名称、事件类型和表达式（函数）。可用函数包括 SUM(DatasetType.COLUMN_NAME) 和 SAMPLECOUNT()。对于 SUM() 函数，指定数据集类型和列名称。例如 SUM(Items.LENGTH)。有关每个参数的信息，请参阅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\"
}]"
```

创建指标归因 (AWS 软件开发工具包)

以下代码演示如何使用 适用于 Python (Boto3) 的 AWS SDK创建指标归因。您指定的角色必须拥有PutMetricData访问您的 Amazon S3 存储桶的 CloudWatch 权限，如果发布到 Amazon S3，则必须拥有访问您的 Amazon S3 存储桶的PutObject权限。要使用您在中创建的角色为 Amazon Personalize 创建 IAM 角色，您可能需要为 CloudWatch 和 Amazon S3 添加策略。有关策略示例，请

参阅让 Amazon Personalize 访问权限 CloudWatch和向 Amazon Personalize 授予访问 Amazon S3 存储桶的权限。

为每个指标指定名称、事件类型和表达式（函数）。可用函数包括 SUM(DatasetType.COLUMN_NAME) 和 SAMPLECOUNT()。对于 SUM() 函数，指定数据集类型和列名称。例如 SUM(Items.LENGTH)。有关每个参数的信息，请参阅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();
```

更新 Amazon Personalize 指标归因

更新指标归因时，您可以添加和移除指标以及修改其输出配置。您可以使用 Amazon Personalize 控制台或 AWS SDK 更新指标归因。 AWS Command Line Interface

主题

- 更新指标归因（控制台）
- 更新指标归因 (AWS CLI)
- 更新指标归因 (AWS SDK)

更新指标归因（控制台）

要使用 Amazon Personalize 控制台更新指标归因，您可以在指标归因页面上进行更改。

更新指标归因

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择您的数据集组。
3. 在导航窗格中，选择指标归因。
4. 在底部，选择指标属性选项卡或指标归因配置选项卡，以开始进行更改。
   - 要添加或删除指标，请选择指标属性选项卡，然后选择编辑属性。在编辑指标属性页面上进行更改，然后选择更新以保存更改。

- 要更改 Amazon S3 输出存储桶或 IAM 服务角色，请选择编辑指标归因配置选项卡，然后在编辑归因配置页面上进行更改。单击更新以保存您的更改。

更新指标归因 (AWS CLI)

创建指标归因后，您可以使用 AWS Command Line Interface (AWS CLI) 添加和移除指标以及修改其输出配置。以下代码显示了如何使用 `update-metric-attribution` 命令删除指标：

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--remove-metrics metricName1 metricName2
```

以下代码显示了如何添加其他指标和指定新的输出配置：

```
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\"
}]"
```

如果成功，则 Amazon Personalize 会返回您更新的指标归因的 ARN。要获得所有参数的完整列表，请参阅 UpdateMetricAttribution。

更新指标归因 (AWS SDK)

创建指标归因后，您可以添加或移除指标以及修改其输出配置。以下代码显示了如何从指标归因中移除指标。

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();
```

以下代码显示了如何添加其他指标和指定新的输出配置：

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();
```

如果成功，则 Amazon Personalize 会返回您更新的指标归因的 ARN。要获得所有参数的完整列表，请参阅 UpdateMetricAttribution。

删除 Amazon Personalize 指标归因

如果您不再需要生成报告，可以删除指标归因。删除指标归因会删除其所有指标和输出配置。

如果您删除指标归因，Amazon Personalize 将停止自动向其发送与之相关的报告 PutEvents 和增量批量数据 CloudWatch。已经发送到 Amazon S3 CloudWatch 或已发布到 Amazon S3 的数据不受影响。您可以使用 Amazon Personalize 控制台或 AWS SDK 删除指标归因。 AWS Command Line Interface

主题

- 删除指标归因（控制台）
- 删除指标归因 (AWS CLI)
- 删除指标归因 (AWS 软件开发工具包)

删除指标归因（控制台）

在指标归因对应的概览页面上删除指标归因。

删除指标归因

1. 打开 Amazon Personalize 控制台https://console.aws.amazon.com/personalize/home并登录您的账户。
2. 选择您的数据集组。
3. 在导航窗格中，选择指标归因。
4. 选择删除，然后确认删除。

删除指标归因 (AWS CLI)

要使用删除指标归因 AWS CLI，请按如下方式使用`delete-metric-attribution`命令。

```
aws personalize delete-metric-attribution --metric-attribution-arn metric attribution
 ARN
```

删除指标归因 (AWS 软件开发工具包)

以下代码演示了如何使用 SDK for Python (Boto3) 删除指标归因：

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();
```

在中查看指标数据的图表 CloudWatch

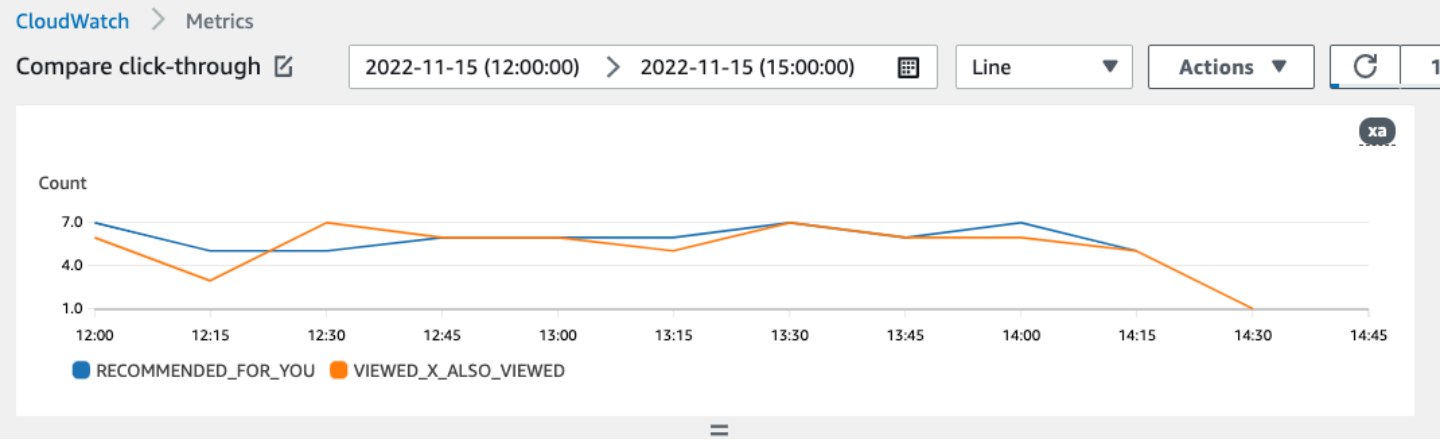
> **Important**
>
> 在创建指标归因并记录事件或导入增量批量数据后，每个指标每月将产生一些 CloudWatch 费用。有关 CloudWatch 定价的信息，请参阅 Amazon CloudWatch 定价页面。要停止向发送指标 CloudWatch，请删除指标归因。

在您创建指标归因后，Amazon Personalize 会自动将来自亚马逊的指标PutEvents和增量批量数据发送给亚马逊 CloudWatch。您可以使用 CloudWatch 控制台选择指标并创建指标数据的图表。这些图表可以帮助您直观地检查和比较不同推荐器或市场活动的效果和影响。

要比较源，每个交互事件都必须包含 `recommendationId` 或 `eventAttributionSource`。有关显示了如何在事件中包括此数据的代码示例，请参阅事件指标和归因报告。

要查看中的指标 CloudWatch，请完成绘制指标图表中的步骤。您可以在不同的详细程度下查看您的数据。您可以绘制图表的最小周期为 15 分钟。您可以查看过去 2 周的 Amazon Personaliz CloudWatch e 数据，较早的数据将被忽略。对于搜索词，指定您在创建指标归因时为该指标提供的名称。

以下是指标可能在中的显示方式的示例 CloudWatch。指标显示两个不同推荐器每 15 分钟的点击率。



将指标归因报告发布到 Amazon S3

对于所有批量数据，如果您在创建指标归因时提供了 Amazon S3 存储桶，则可以选择在每次为交互数据创建数据集导入作业时，将指标报告发布到您的 Amazon S3 存储桶。

要将指标发布到 Amazon S3，您需要在指标归因中提供指向 Amazon S3 存储桶的路径。然后，在创建数据集导入作业时，将报告发布到 Amazon S3。作业完成后，可以在 Amazon S3 存储桶中找到指

标。每次发布指标时，Amazon Personalize 都会在您的 Amazon S3 存储桶中创建一个新文件。文件名包括导入方法和日期，如下所示：

`AggregatedAttributionMetrics - ImportMethod - Timestamp.csv`

以下是指标报告 CSV 文件前几行显示方式的示例。此示例中的指标报告两个不同推荐器在 15 分钟间隔内的总单击量。在 EVENT_ATTRIBUTION_SOURCE 列中，每个推荐器由其 Amazon 资源名称 (ARN) 标识。

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

将批量数据的指标发布到 Amazon S3（控制台）

要使用 Amazon Personalize 控制台将指标发布到 Amazon S3 存储桶，请创建数据集导入作业，然后在将事件指标发布到 S3 中选择发布此导入作业的指标。

如需分步指导，请参阅 创建数据集导入作业（控制台）。

将批量数据的指标发布到 Amazon S3 (AWS CLI)

要使用 AWS Command Line Interface (AWS CLI) 将指标发布到 Amazon S3 存储桶，请使用以下代码创建数据集导入任务并提供标`publishAttributionMetricsToS3`志。如果您不想发布特定作业的指标，请忽略标志。有关每个参数的信息，请参阅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
```

将批量数据的指标发布到 Amazon S3 (AWS 软件开发工具包)

要使用软件开发工具包将指标发布到 Amazon AWS S3 存储桶，请创建数据集导入任务并将其设置为 publishAttributionMetricsToS3 true。有关每个参数的信息，请参阅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();
```

通过 A/B 测试衡量推荐的影响

执行 A/B 测试包括运行具有多种变体的实验并比较结果。使用 Amazon Personalize 推荐进行 A/B 测试需要向不同的用户群展示不同类型的推荐，然后比较结果。您可以使用 A/B 测试来帮助比较和评估不同的推荐策略，并衡量建议的影响。

例如，您可以使用 A/B 测试来查看 Amazon Personalize 推荐是否提高了点击率。要测试该场景，您可以向一组用户展示非个性化的建议，例如特色产品。您可以向另一组展示由 Amazon Personalize 生成的个性化建议。当您的客户与物品交互时，您可以记录结果，并查看哪种策略的点击率最高。

使用 Amazon Personalize 推荐进行 A/B 测试的工作流程如下：

1. 计划您的实验 - 定义可量化的假设，确定业务目标，定义实验变体，并确定实验时间框架。
2. 拆分用户 - 将用户分成两组或多个组，包括一个对照组和一个或多个实验组。
3. 运行您的实验 - 向实验组中的用户显示修改后的建议。向对照组中的用户显示未做任何更改的建议。记录他们与建议的交互以跟踪结果。
4. 评估结果 - 分析实验结果，以确定修改是否对实验组产生了统计学上的显著差异。

主题

- A/B 测试最佳实践

A/B 测试最佳实践

使用以下最佳实践来帮助您设计和维护针对 Amazon Personalize 推荐的 A/B 测试。

- 确定可量化的业务目标。验证您要比较的不同建议是否与此业务目标一致，并且与不同的或不可量化的目标无关。

- 定义与您的业务目标一致的可量化的假设。例如，您可能会预测，您自己的定制内容的推广将使这些物品的点击量增加 20%。您的假设决定了您对实验组所做的修改。
- 定义与您的假设相关的关键绩效指标 (KPIs)。你 KPIs 用来衡量实验的结果。这些可能是以下内容：
  - 点击率
  - 观看时间
  - 总价格
- 根据您的假设，验证实验中的用户总数是否够大，足以得出统计学上显著的结果。
- 在开始实验之前，定义流量分流策略。避免在实验运行时更改流量分流。
- 除了与实验相关的修改（例如模型）外，实验组和对照组的应用程序或网站的用户体验应保持一致。用户体验的变化（例如 UI 或延迟）可能会产生误导性结果。
- 控制外部因素，例如节假日、持续的营销活动和浏览器限制。这些外部因素可能会产生误导性结果。
- 除非与您的假设或业务需求直接相关，否则避免更改 Amazon Personalize 建议。诸如应用筛选器或手动更改顺序之类的更改可能会产生误导性结果。
- 评估结果时，确保在得出结论之前结果具有统计学意义。行业标准的显著性水平为 5%。

对 OpenSearch 搜索结果进行个性化设置

您可以使用 Amazon Personalize 为您的用户对开源 OpenSearch 或 Amazon OpenSearch Service 结果进行个性化设置。

OpenSearch 是一项基于 Apache 2.0 许可证的自托管开源搜索服务。Amazon OpenSearch Service 是一项托管服务，可帮助您在 AWS 云中部署、运行并扩展 OpenSearch 资源。当您使用 Amazon OpenSearch Service 时，OpenSearch 会检索结果并对其进行排名。

对查询结果排名时，OpenSearch 使用名为 BM-25 的概率排名框架来计算相关性得分。如果独特的关键词在文档中出现较为频繁，BM-25 会为该文档分配更高的相关性分数。OpenSearch 排名不考虑点击数据等用户行为。

将 Amazon Personalize 与 OpenSearch 结合使用时，Amazon Personalize 会根据用户过去的行为、有关物品的任何元数据以及用户相关的任何元数据，对 OpenSearch 结果进行重新排名。随后，OpenSearch 会合并重新排名，然后再将搜索响应返回到您的应用程序。在将 Amazon Personalize 的排名应用于 OpenSearch 结果时，您可以控制 OpenSearch 给予该排名的权重。

通过这个重新排名，结果可以更具吸引力，更符合用户的兴趣。这可以提高应用程序的点击率和转化率。有关描述个性化搜索如何改善电子商务应用程序结果的使用案例示例，请参阅使用案例示例。

在开始对 OpenSearch 结果进行个性化设置之前，请查看Amazon Personalize 搜索排名插件要求中列出的要求。

主题

- 使用案例示例
- Amazon Personalize 搜索排名插件的工作原理
- 其他信息
- Amazon Personalize 搜索排名插件要求
- 使用 Amazon Personalize 对来自亚马逊 OpenSearch 服务的结果进行个性化设置
- 使用 Amazon Personalize 对开源 OpenSearch 的结果进行个性化设置
- personalized_search_ranking 响应处理器的字段
- 管道指标示例

使用案例示例

当您使用 Amazon Personalize 对 OpenSearch 结果进行重新排名时，搜索结果可能与您的用户更加相关。例如，您可能有一个销售汽车的电子商务应用程序。如果您的用户输入了丰田汽车的查询，而您没有对结果进行个性化设置，则 OpenSearch 将根据您数据中的关键字返回丰田汽车制造的汽车列表。对于所有用户，此列表将按相同的顺序进行排名。

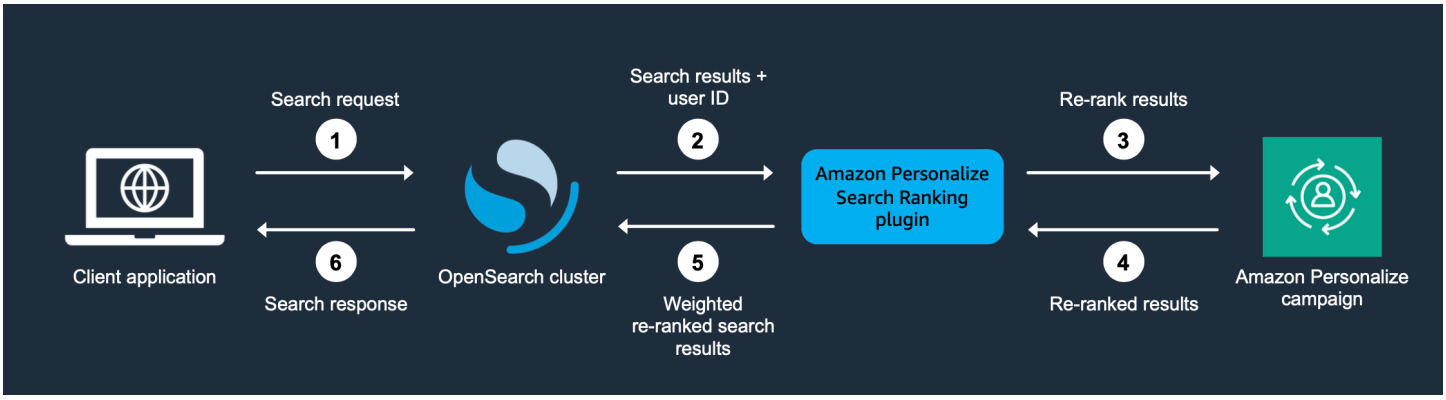
但是，如果您使用 Amazon Personalize 对结果进行个性化设置，则 OpenSearch 会根据特定用户的行为（例如单击次数）按与特定用户的相关性顺序对这些汽车进行重新排名。用户最有可能单击的汽车排在第一位。

在对 OpenSearch 结果进行个性化设置时，您可以控制 OpenSearch 给予 Amazon Personalize 排名的权重（重视度）。继续此示例，如果用户搜索特定年份特定类型的汽车（例如 2008 Toyota Prius），则您可能需要更加重视 OpenSearch 的原始排名。

但是，对于能够产生广泛结果且更通用的查询（例如搜索所有丰田汽车），您可能会高度重视个性化。这样，列表顶部的汽车与特定用户更加相关。

Amazon Personalize 搜索排名插件的工作原理

下图显示了 Amazon Personalize 搜索排名插件的工作原理。



1. 您将客户的查询提交到您的 OpenSearch Service 域或您的开源 OpenSearch 集群。
2. OpenSearch 将查询响应（与查询相关的物品列表）和用户 ID 发送到 Amazon Personalize 搜索排名插件。
3. 该插件将响应中的物品和用户发送到您的 Amazon Personalize 市场活动，以进行排名。它使用搜索管道中的食谱和市场活动 Amazon 资源名称 (ARN) 值为用户获取个性化排名。它使用 GetPersonalizedRanking API 操作来获取建议。在请求中，它传递进行查询的用户的 userId 以及从 `inputList` 中 OpenSearch 查询返回的项。

4. Amazon Personalize 将重新排名的结果返回给插件。
5. 该插件重新排列搜索结果并将其返回到您的 OpenSearch Service 域或开源 OpenSearch 集群。它根据 Amazon Personalize 市场活动的响应以及您在设置过程中指定的个性化重视程度，对结果重新排名。
6. 您的开源 OpenSearch 集群或 OpenSearch Service 域将最终结果返回到您的应用程序。

其他信息

以下资源提供有关使用 OpenSearch 的更多信息。

- 有关开源 OpenSearch 入门的信息，请参阅快速入门。
- 有关 OpenSearch Service 入门的信息，请参阅《Amazon OpenSearch Service 开发人员指南》中的 Amazon OpenSearch Service 入门。
- 有关 Amazon Personalize 中的 Personalized-Ranking 配方的信息，请参阅 Personalized-Ranking-v2 配方或 Personalized-Ranking 食谱。

Amazon Personalize 搜索排名插件要求

在开始对 OpenSearch 结果进行个性化设置之前，注意 Amazon Personalize 搜索排名插件的以下指南和要求：

- 您必须使用 OpenSearch 2.9.0 或更高版本。如果您使用 Amazon OpenSearch Service，则您的域必须使用 2.9 或更高版本。
- 如果您还没有，请完成设置权限中的说明，向您的用户授予访问 Amazon Personalize 的权限，并向 Amazon Personalize 授予访问您在 Amazon Personalize 中资源的权限。
- 您必须能够从 OpenSearch Service 域或开源 OpenSearch 集群访问您的 Amazon Personalize 资源。
  - 有关授予 OpenSearch Service 域访问权限的信息，请参阅设置亚马逊 OpenSearch 服务权限。
  - 有关授予 OpenSearch 集群访问权限的信息，请参阅设置开源 OpenSearch 权限。
- 您只能使用自定义 Amazon Personalize 资源。如果您创建了域数据集组，则仍然可以添加自定义资源。
- 您只能使用自定义配方 Personalized-Ranking-v2 配方 或 Personalized-Ranking 食谱。
- 您必须在 Amazon Personalize 中创建物品交互数据集。物品和用户数据集是可选项。

- 使用 Amazon Personalize 搜索排名插件时，您无法应用 Amazon Personalize 筛选器。
- 默认情况下，该插件假定 OpenSearch 中已编入索引的文档 `_id` 与 Amazon Personalize 数据中的 itemId 相匹配。如果 OpenSearch 数据使用与 Amazon Personalize itemId 相对应的其他字段，则您在配置插件时必须指定该字段的名称。
- 您用于进行查询的用户的 userId 必须与您导入到 Amazon Personalize 的数据中的 userId 相匹配。
- 该插件仅对来自 OpenSearch 的前 500 个搜索结果重新排名。不会对其余项重新排名，这些项最终排在列表底部。

使用 Amazon Personalize 对来自亚马逊 OpenSearch 服务的结果进行个性化设置

要对来自亚马逊 OpenSearch 服务的 OpenSearch 结果进行个性化设置，请执行以下操作：

1. 设置 Amazon Personalize - 如果您尚未设置，请完成设置 Amazon Personalize 中的步骤来设置凭证，并为 Amazon Personalize 设置权限。您无需设置即可对 OpenSearch 结果 AWS SDKs 进行个性化设置。
2. 完成 Amazon Personalize 工作流程 — 完成 Amazon Personalize 工作流程以导入数据，使用 Personalized-Ranking-v 2 级或个性化排名方法创建解决方案，训练自定义解决方案版本，然后将其部署到活动中。您必须创建物品交互数据集。用户数据集和物品数据集是可选项。有关更多信息，请参阅 Amazon Personalize 工作流程详细信息。
3. 设置权限-设置权限，以便您可以从 OpenSearch 服务域访问您的 Amazon Personalize 资源。有关更多信息，请参阅 设置权限。
4. 安装 Amazon Personalize 搜索排名插件 - 此插件可处理与 Amazon Personalize 的通信并对结果重新排名。有关在 OpenSearch 服务域上安装插件的信息，请参阅安装插件。
5. 配置 Amazon Personalize 搜索排名插件 - 要配置该插件，您需要创建搜索管道。搜索管道 是一组请求和响应处理器。为插件创建管道时，需在 `personalized_search_ranking` 响应处理器中指定您的 Amazon Personalize 资源。此外，还要配置插件在重新对结果排名时对 Amazon Personalize 结果给予的权重。有关更多信息，请参阅 创建管道。
6. 将 Amazon Personalize Search Ranking 插件应用于 OpenSearch 查询 — 在使用`personalized_search_ranking`响应处理器配置搜索渠道后，您就可以将 Amazon Personalize 搜索排名插件应用于您的 OpenSearch 查询并查看重新排名的结果了。有关将插件应用于 OpenSearch 服务查询的信息，请参阅应用插件。
7. 比较结果 — Amazon Personalize 搜索排名插件会对 OpenSearch查询响应中的搜索结果进行重新排名。它既考虑了 Amazon Personalize 的排名，也考虑了来自的排名 OpenSearch。要了解如何对

结果进行重新排名，您可以比较使用个性化的查询和不使用个性化的查询的结果。有关将结果与 S OpenSearch ervice 进行比较的信息，请参见比较结果。

8. 监控 Amazon Personalize 搜索排名插件 - 将 Amazon Personalize 搜索排名插件应用于搜索查询时，您可以通过获取搜索渠道的指标来监控该插件。有关使用 S OpenSearch ervice 监控插件的信息，请参阅监控插件。

主题

- 设置亚马逊 OpenSearch 服务权限
- 在 OpenSearch Service 域上安装 Amazon Personalize 搜索排名插件
- 在 Amazon OpenSearch Service 中创建管道
- 将插件应用于 Amazon OpenSearch Service 查询
- 比较个性化 Amazon OpenSearch Service 结果与不进行个性化的结果
- 通过 Amazon OpenSearch Service 监控插件

设置亚马逊 OpenSearch 服务权限

如果您使用亚马逊 OpenSearch 服务，则必须能够从您的 OpenSearch 服务域访问您的 Amazon Personalize 资源。

设置 权限

1. 根据您的资源是在相同账户还是不同账户中，创建一个或多个有权访问资源的 IAM 服务角色。
   - 如果您的 OpenSearch 服务和 Amazon Personalize 资源位于同一个账户中，则您可以为服务创建 IAM 服务角色，并授予其从您的 Amazon P OpenSearch ersonalize 活动中获得个性化排名的权限。有关更多信息，请参阅 当资源位于同一账户时配置权限。
   - 如果您的 OpenSearch 服务和 Amazon Personalize 资源位于不同的账户中，则可以创建两个 IAM 服务角色。您可以使用您的 OpenSearch 服务资源在账户中创建一个账户，并授予其访问您的 OpenSearch 服务资源的权限。然后在 Amazon Personalize 资源账户中创建一个角色，并向其授予从 Amazon Personalize 市场活动中获得个性化排名的权限。有关更多信息，请参阅 当资源位于不同账户时配置权限。
2. 向访问您的 OpenSearch 服务域的用户或角色授予您为服务创建的 IAM OpenSearch 服务角色的`PassRole`权限。有关更多信息，请参阅 配置亚马逊 OpenSearch 服务域安全。

设置权限之后，就可以将插件安装到您的域中。有关更多信息，请参阅 安装插件。

主题

- 当资源位于同一账户时配置权限
- 当资源位于不同账户时配置权限
- 配置亚马逊 OpenSearch 服务域安全

当资源位于同一账户时配置权限

如果您的 OpenSearch 服务和 Amazon Personalize 资源位于同一个账户中，则必须为服务创建 IAM OpenSearch 服务角色。此角色必须拥有权限才能从 Amazon Personalize 市场活动中获得个性化排名。要向您的 OpenSearch 服务服务角色授予权限，才能从您的 Amazon Personalize 活动中获得个性化排名，需要满足以下条件：

- 角色的信任策略必须授予 OpenSearch 服务`AssumeRole`权限。有关信任策略示例，请参阅信任策略示例。
- 此角色必须拥有权限才能从 Amazon Personalize 市场活动中获得个性化排名。有关策略示例，请参阅权限策略示例。

有关创建 IAM 角色的信息，请参阅《IAM 用户指南》中的创建 IAM 角色。有关将 IAM 策略附加到角色的信息，请参阅《IAM 用户指南》中的添加和删除 IAM 身份权限。

为服务创建 IAM OpenSearch 服务角色后，您必须向访问您的 OpenSearch 服务域的用户或角色授予该 OpenSearch 服务角色的`PassRole`权限。有关更多信息，请参阅 配置亚马逊 OpenSearch 服务域安全。

主题

- 信任策略示例
- 权限策略示例

信任策略示例

以下信任策略示例授予 OpenSearch 服务`AssumeRole`权限。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Principal": {
            "Service": [
                "es.amazonaws.com"
            ]
        }
    }]
}
```

权限策略示例

以下策略示例向该角色授予从 Amazon Personalize 市场活动中获得个性化排名的最低权限。对于 `Campaign ARN`，为 Amazon Personalize 市场活动指定 Amazon 资源名称 (ARN)。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

当资源位于不同账户时配置权限

如果您的 S OpenSearch ervice 和 Amazon Personalize 资源位于不同的账户中，则您可以在每个账户中创建一个 IAM 角色并向该角色授予对账户中资源的访问权限。

设置多个账户的权限

1. 在 Amazon Personalize 市场活动所在的账户中，创建一个有权从 Amazon Personalize 市场活动中获取个性化排名的 IAM 角色。配置插件时，可以在 `personalized_search_ranking` 响应处理器的 `external_account_iam_role_arn` 参数中指定此角色的 ARN。有关更多信息，请参阅 在 Amazon OpenSearch Service 中创建管道。

   有关策略示例，请参阅权限策略示例。

2. 在您的 OpenSearch 服务域所在的账户中，创建一个具有授予 OpenSearch 服务`AssumeRole`权限的信任策略的角色。配置插件时，可以在 `personalized_search_ranking` 响应处理器的 `iam_role_arn` 参数中指定此角色的 ARN。有关更多信息，请参阅 在 Amazon OpenSearch Service 中创建管道。

   有关信任策略示例，请参阅信任策略示例。

3. 修改每个角色以向其他角色授予 `AssumeRole` 权限。例如，对于有权访问您的 Amazon Personalize 资源的角色，其 IAM 策略将向账户中具有 OpenSearch 服务域的角色授予担任角色的权限，如下所示：

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [{
           "Sid": "",
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::111122223333:role/roleName"

       }]
   }
   ```

4. 在您的 OpenSearch 服务域所在的账户中，向访问您的 OpenSearch 服务域的用户或角色授予您刚刚创建的 OpenSearch 服务角色的`PassRole`权限。有关更多信息，请参阅 配置亚马逊 OpenSearch 服务域安全。

配置亚马逊 OpenSearch 服务域安全

要将插件与 S OpenSearch ervice 配合使用，访问您域的用户或角色必须拥有您刚刚创建的 Service 的 IAM OpenSearch 服务角色的`PassRole`权限。此外，此用户或角色必须具有执行 `es:ESHttpGet` 和 `es:ESHttpPut` 操作的权限。

有关配置 OpenSearch 服务访问权限的信息，请参阅《亚马逊 OpenSearch 服务开发者指南》中的 “亚马逊 OpenSearch 服务安全”。有关策略示例，请参阅 OpenSearch 服务用户或角色的策略示例。

OpenSearch 服务用户或角色的策略示例

以下 IAM 策略示例向用户或角色`PassRole`授予您为服务创建的 IAM OpenSearch 服务角色的权限当资源位于同一账户时配置权限。

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"
        }
    ]
}
```

以下 IAM 策略授予使用 S OpenSearch ervice 创建管道和搜索查询的最低权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:ESHttpGet",
```

```
                "es:ESHttpPut"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:ResourceTag/environment": [
                        "production"
                    ]
                }
            }
        }
    ]
}
```

在 OpenSearch Service 域上安装 Amazon Personalize 搜索排名插件

完成 Amazon Personalize 工作流程并满足 插件要求 中列出的要求后，就可以在域中安装插件。

要使用插件，请将 `Amazon_Personalize_Search_Ranking_Plugin` 插件与域相关联。该插件已预装，您无需从 Amazon S3 将其导入。关联插件的方式与关联 OpenSearch Service 包的方式相同。有关关联 OpenSearch Service 包的信息，请参阅 Amazon OpenSearch Service 的自定义包。

将插件与域关联后，就可以配置该插件。要配置该插件，需创建搜索管道并指定 `personalized_search_ranking` 响应处理器。有关更多信息，请参阅 创建管道。

有关 Amazon OpenSearch Service 域的其他信息

以下资源提供有关使用 Amazon OpenSearch Service 域的其他信息。

- 有关配置测试域的简明教程，请参阅《Amazon OpenSearch Service 开发人员指南》中的步骤 1：创建 Amazon OpenSearch Service 域。
- 有关配置 OpenSearch Service 域的更详细步骤，请参阅创建和管理 Amazon OpenSearch Service 域。
- 有关将少量测试数据上传到 OpenSearch Service 的简明教程，请参阅《Amazon OpenSearch Service 开发人员指南》中的步骤 2：将数据上传到 Amazon OpenSearch Service 以编入索引。
- 有关摄取数据的完整信息，请参阅《Amazon OpenSearch Service 开发人员指南》中的将 Amazon OpenSearch Service 中的数据编入索引。

在 Amazon OpenSearch Service 中创建管道

安装 Amazon Personalize 搜索排名插件后，就可以对其进行配置了，方法是：创建 OpenSearch 搜索管道。

搜索管道 是一组请求和响应处理器，可按照您创建此类处理器的顺序依次对其运行。为插件创建搜索管道时，指定 personalized_search_ranking 响应处理器。有关搜索管道的信息，请参阅搜索管道。

创建带有 personalized_search_ranking 响应处理器的搜索管道后，就可以开始将该插件应用于 OpenSearch 查询了。您可以将其应用于 OpenSearch 索引或单个 OpenSearch 查询。有关更多信息，请参阅 应用插件。

您可以使用以下 Python 代码在 OpenSearch Service 域上创建带有 personalized_search_ranking 响应处理器的搜索管道。将 domain endpoint 替换为您的域端点 URL。例如：https://<domain name>.<AWS region>.es-staging.amazonaws.com。有关每个 personalized_search_ranking 参数的完整说明，请参阅 personalized_search_ranking 响应处理器的字段。

```
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}")
```

将插件应用于 Amazon OpenSearch Service 查询

创建管道后，就可以将 Amazon Personalize 搜索排名插件应用于查询。您可以将 Amazon Personalize 搜索排名插件应用于索引的所有查询和响应。您也可以将该插件应用于单个查询和响应。

- 您可以使用以下 Python 代码将搜索管道应用于索引。通过这种方法，所有使用此索引的搜索都使用该插件将个性化应用于搜索结果。

```
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}")
```

- 您可以使用以下 Python 代码，将搜索管道应用于针对丰田品牌汽车的单个查询。

  更新代码，以指定域端点、OpenSearch Service 索引、管道名称和查询。对于 `user_id`，指定要获取搜索结果的用户的 ID。该用户必须出现在您用于创建 Amazon Personalize 解决方案版本的数据中。如果用户不在此类数据中，则 Amazon Personalize 会根据物品的受欢迎程度对其进行排名。

  对于 `context`，如果您使用上下文元数据，请提供用户的上下文元数据，例如其设备类型。`context` 字段为可选项。有关更多信息，请参阅 利用上下文元数据提高建议的相关性。

```
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}")
```

比较个性化 Amazon OpenSearch Service 结果与不进行个性化的结果

Amazon Personalize 搜索排名插件根据 Amazon Personalize 的排名和 OpenSearch 的排名，重新排列搜索结果。插件对结果进行重新排名的方式取决于您在管道中配置 `personalized_search_ranking` 响应处理器的方式。

要了解如何对结果进行排名，您可以在设置个性化和不设置个性化的情况下运行查询，然后比较结果。您可以使用以下 Python 代码来运行两个不同的查询，并将结果输出到两个 JSON 文件中。第一种方法运行一个使用插件对结果进行重新排名的查询。第二种方法运行一种不设置个性化时生成结果的方法。

```
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)
```

通过 Amazon OpenSearch Service 监控插件

将 Amazon Personalize 搜索排名插件应用于 OpenSearch 查询时，您可以通过获取搜索管道的指标，监控插件。管道指标包括诸如 `personalized_search_ranking` 响应处理器的失败请求数量之类的统计信息。

如果您使用 OpenSearch Service，则可以通过 Amazon CloudWatch 中的指标监控插件。有关更多信息，请参阅监控 Amazon OpenSearch Service 域。

您可以使用以下 Python 代码获取所有管道的指标。有关管道指标的示例，请参阅管道指标示例。

```
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}")
```

使用 Amazon Personalize 对开源 OpenSearch 的结果进行个性化设置

要对开源结果进行个性化设置 OpenSearch，请执行以下操作：

1. 设置 Amazon Personalize - 如果您尚未设置，请完成设置 Amazon Personalize 中的步骤来设置凭证，并为 Amazon Personalize 设置权限。您无需设置即可对 OpenSearch 结果 AWS SDKs 进行个性化设置。
2. 完成 Amazon Personalize 工作流程 — 完成 Amazon Personalize 工作流程以导入数据，使用 Personalized-Ranking-v 2 级或个性化排名食谱创建解决方案，训练自定义解决方案版本，然后将其部署到活动中。您必须创建物品交互数据集。用户数据集和物品数据集是可选项。有关更多信息，请参阅 Amazon Personalize 工作流程详细信息。
3. 设置权限-设置权限，以便您可以从 OpenSearch 集群访问您的 Amazon Personalize 资源。有关更多信息，请参阅 设置权限。

4. 安装 Amazon Personalize 搜索排名插件 - Amazon Personalize 搜索排名插件可处理与 Amazon Personalize 的通信并对结果重新排名。
   - 如果您已经有一个正在运行的 OpenSearch 集群，则可以手动安装该插件。有关更多信息，请参阅 在现有的 OpenSearch 集群上手动安装插件。
   - 如果您尚未创建 OpenSearch 集群，则可以使用快速入门 bash 脚本来创建集群。有关更多信息，请参阅 使用脚本创建集群和安装插件。
5. 配置 Amazon Personalize 搜索排名插件 - 要配置该插件，您需要创建搜索管道。搜索管道 是一组请求和响应处理器。为插件创建管道时，需在 `personalized_search_ranking` 响应处理器中指定您的 Amazon Personalize 资源。此外，还要配置插件在重新对结果排名时对 Amazon Personalize 结果给予的权重。有关更多信息，请参阅 创建管道。
6. 将 Amazon Personalize 搜索排名插件应用于 OpenSearch 查询 — 您可以将 Amazon Personalize 搜索排名插件应用于 OpenSearch 索引的所有查询和回复。您也可以将该插件应用于单个 OpenSearch 查询和响应。有关将插件应用于开源查询的信息 OpenSearch，请参阅应用插件。
7. 比较结果 — Amazon Personalize 搜索排名插件会对 OpenSearch查询响应中的搜索结果进行重新排名。它既考虑了 Amazon Personalize 的排名，也考虑了来自的排名 OpenSearch。要了解如何对结果进行重新排名，您可以比较使用个性化的查询和不使用个性化的查询的结果。有关将结果与开源进行比较的信息 OpenSearch，请参阅比较结果。
8. 监控 Amazon Personalize 搜索排名插件 - 将 Amazon Personalize 搜索排名插件应用于搜索查询时，您可以通过获取搜索渠道的指标来监控该插件。有关在开源 OpenSearch 集群上监控插件的信息，请参阅通过开源 OpenSearch 监控插件。有关从中返回的管道指标的摘录 OpenSearch，请参阅管道指标示例。

主题

- 设置开源 OpenSearch 权限
- 在现有的 OpenSearch 集群上手动安装 Amazon Personalize 搜索排名插件
- 使用脚本创建集群和安装插件
- 在开源 OpenSearch 中创建管道
- 将 Amazon Personalize 搜索排名插件应用于开源 OpenSearch 中的查询
- 比较个性化 OpenSearch 结果与不进行个性化的结果
- 通过开源 OpenSearch 监控插件

设置开源 OpenSearch 权限

如果您使用开源 OpenSearch，则必须能够从您的开放搜索集群访问您的 Amazon Personalize 资源。要授予访问权限，请执行以下操作：

- 如果您是 OpenSearch 从头开始设置，则可以使用快速启动 bash 脚本在 Docker 容器中运行 OpenSearch 集群。该脚本使用您的 AWS 个人资料中的默认凭据。您可以在运行该脚本时指定备用配置文件。

  这些证书必须与有权为您的 Amazon Personalize 活动执行 GetPersonalizedRanking操作的用户或角色相关联。如需 IAM 策略示例，请参阅 IAM 策略示例。或者，此类凭证必须有权代入具有这些权限的角色。您可以在为 Amazon Personalize 搜索排名插件创建管道时，为此角色提供 Amazon 资源名称 (ARN)。
- 如果您不使用快速启动 bash 脚本，则可以手动将凭据添加到 OpenSearch 密钥库中。这些证书必须与有权为您的 Amazon Personalize 活动执行 GetPersonalizedRanking 操作的用户或角色相对应。

  要手动将您的 AWS 凭据添加到 OpenSearch 密钥库中，请在 OpenSearch 集群运行的地方（例如 Docker 容器）运行以下命令。然后，提供每个凭证。如果您不使用会话令牌，则可以省略命令中的最后一行。

  ```
  opensearch-keystore add \
  personalized_search_ranking.aws.access_key \
  personalized_search_ranking.aws.secret_key \
  personalized_search_ranking.aws.session_token
  ```
- 如果您在 Amazon EC2 实例上运行 OpenSearch 集群，则可以使用 IAM 实例配置文件授予权限。该角色所附的政策必须授予其在您的 Amazon Personalize 活动中执行 GetPersonalizedRanking 操作的权限。它还必须向 Amazon EC2 授予代入角色的权限。

  有关 Amazon EC2 实例配置文件的信息，请参阅使用实例配置文件。有关策略示例，请参阅IAM 策略示例。

IAM 策略示例

以下策略示例向用户或角色授予从 Amazon Personalize 市场活动中获得个性化排名的最低权限。对于 `Campaign ARN`，为 Amazon Personalize 市场活动指定 Amazon 资源名称 (ARN)。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:GetPersonalizedRanking"
            ],
            "Resource": "arn:aws:personalize:us-
east-1:111122223333:campaign/YourResourceId"
        }
    ]
}
```

此外，如果您在 Amazon EC2 实例上运行 OpenSearch 集群并使用 IAM 实例配置文件授予权限，则该角色的信任策略必须按如下方式授予 Amazon EC2 `AssumeRole` 权限。有关 Amazon EC2 实例配置文件的信息，请参阅使用实例配置文件。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

在现有的 OpenSearch 集群上手动安装 Amazon Personalize 搜索排名插件

如果您已经有一个 OpenSearch 集群，则可以直接从 OpenSearch GitHub 存储库在集群上手动安装该插件。

手动安装插件

1. 使用以下命令启动 OpenSearch 集群：

   ```
   bin/opensearch
   ```

2. 将目录数据上传到 OpenSearch 集群（如果尚未执行此操作）。上传数据时，创建 OpenSearch 索引并定义字段映射。然后，将数据上传到该索引。有关示例，请参阅使用示例数据创建索引和字段映射。
3. 使用以下命令安装插件：

   ```
   bin/opensearch-plugin install https://github.com/opensearch-project/search-processor/releases/download/2.9.0/opensearch-search-processor-2.9.0.0.zip
   ```

   有关安装插件的更多信息，请参阅安装插件。

安装 Amazon Personalize 搜索排名插件后，就可以配置该插件了。要配置该插件，需创建搜索管道并指定 `personalized_search_ranking` 响应处理器。有关更多信息，请参阅 创建管道。

使用脚本创建集群和安装插件

如果您尚未创建 OpenSearch 集群，则可以使用快速入门 bash 脚本创建一个。此脚本在 Docker 容器中设置 OpenSearch 集群，使用您的默认 AWS 配置文件设置凭证，并安装 Amazon Personalize 搜索排名插件。

有关手动创建 OpenSearch 集群的信息，请参阅 OpenSearch 文档中的快速入门说明。

使用快速入门 bash 脚本安装插件

1. 在运行脚本之前，下载并安装适用于您的操作系统的 Docker 桌面。
2. 从 GitHub 下载快速入门 bash 脚本。
3. 在工作目录中，使用以下命令运行该脚本。

```
sh personalized_search_ranking_quickstart.sh
```

通过此命令，该脚本使用您的默认 AWS 配置文件中的凭证。要提供备用配置文件，请使用 `--profile` 参数。

```
sh personalized_search_ranking_quickstart.sh --profile profile-name
```

运行脚本后，您可以在 README 文件中找到有关该脚本的更多信息，该文件位于该脚本创建的唯一目录中。此目录存储脚本使用的 Dockerfile 和 docker-compose.yml 文件。例如：`../opensearch-personalize-intelligent-ranking-docker.1234/README`。

4. 将您的目录数据上传到您的 OpenSearch 集群。上传数据时，创建 OpenSearch 索引并定义字段映射。然后，将数据上传到该索引。有关示例，请参阅使用示例数据创建索引和字段映射。

设置 OpenSearch 并安装 Amazon Personalize 搜索排名插件后，就可以对其进行配置了。要配置该插件，需创建搜索管道并指定 `personalized_search_ranking` 响应处理器。有关更多信息，请参阅创建管道。

在开源 OpenSearch 中创建管道

在集群上安装插件之后，就可以对其进行配置了，方法是：创建 OpenSearch 搜索管道。

搜索管道 是一组请求和响应处理器，可按照您创建此类处理器的顺序依次对其运行。为插件创建搜索管道时，指定 `personalized_search_ranking` 响应处理器。有关搜索管道的信息，请参阅搜索管道。

创建带有 `personalized_search_ranking` 响应处理器的管道后，就可以开始将该插件应用于查询。有关更多信息，请参阅 应用插件。

您可以使用以下 curl 命令在开源 OpenSearch Service 集群上创建带有 `personalized_search_ranking` 响应处理器的搜索管道。有关每个 `personalized_search_ranking` 参数的完整说明，请参阅 personalized_search_ranking 响应处理器的字段。

```
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
      }
    }
  ]
}'
```

创建带有 `personalized_search_ranking` 响应处理器的搜索管道后，就可以开始将该插件应用于 OpenSearch 查询了。您可以将其应用于 OpenSearch 索引或单个 OpenSearch 查询。有关更多信息，请参阅 将 Amazon Personalize 搜索排名插件应用于开源 OpenSearch 中的查询。

将 Amazon Personalize 搜索排名插件应用于开源 OpenSearch 中的查询

您可以将 Amazon Personalize 搜索排名插件应用于 OpenSearch 索引的所有查询和响应。您也可以将该插件应用于单个 OpenSearch 查询和响应。

- 以下 curl 命令将搜索管道应用于本地运行的开源 OpenSearch 集群中的 OpenSearch 索引。通过这种方法，所有使用此索引的搜索都使用该插件将个性化应用于搜索结果。

  ```
  curl -XGET "https://localhost:9200/index/_settings" -ku 'admin:admin' --insecure -H
   'Content-Type: application/json' -d'
  {
    "index.search.default_pipeline": "pipeline-name"
  }
  '
  ```

- 以下 curl 命令将搜索管道应用于本地运行的开源 OpenSearch 集群索引中针对丰田品牌汽车的单个查询。

  对于 `user_id`，指定要获取搜索结果的用户的 ID。该用户必须出现在您用于创建 Amazon Personalize 解决方案版本的数据中。如果用户不在此类数据中，则 Amazon Personalize 会根据物品的受欢迎程度对其进行排名。对于 `context`，如果您使用上下文元数据，请提供用户的上下文元

数据，例如其设备类型。context 字段为可选项。有关更多信息，请参阅 利用上下文元数据提高建议的相关性。

```
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" }
    }
  }
}
'
```

要了解如何对结果进行重新排名，您可以通过该插件使用 OpenSearch 控制面板，将 OpenSearch 结果与重新排名的结果进行比较。有关更多信息，请参阅 比较个性化 OpenSearch 结果与不进行个性化的结果。

将插件应用于 OpenSearch 查询时，您可以通过获取 OpenSearch 管道的指标来监控插件。有关更多信息，请参阅 通过开源 OpenSearch 监控插件。

比较个性化 OpenSearch 结果与不进行个性化的结果

要了解如何对结果进行重新排名，您可以在两个单独的浏览器窗口中使用开发工具控制台运行查询。然后，您可以比较个性化设置和不使用个性化设置的查询结果。

使用开发工具控制台比较结果

1. 确保已安装 OpenSearch 控制面板。快速入门 bash 脚本会安装 OpenSearch 控制面板。如果您不使用该脚本或已有集群在运行，则必须安装 OpenSearch 控制面板。有关更多信息，请参阅安装 OpenSearch 控制面板。
2. 启动 OpenSearch 控制面板 使用浏览器打开 http://localhost:5601，并登录 OpenSearch 控制面板。默认凭证是用户名“admin”和密码“admin”。

3. 在 OpenSearch 控制面板主页的“管理”菜单下，选择开发工具。
4. 打开一个单独的浏览器窗口，然后再次打开开发工具控制台。您可以使用前一个窗口中的 URL。
5. 在一个窗口中，输入查询，以不使用任何个性化重新排名。在另一个窗口中，输入 curl 命令，以使用带有 `personalized_search_ranking` 响应处理器的管道。如果您将 curl 命令直接粘贴到控制台中，则该命令会自动转换为控制台使用的格式。有关命令示例，请参阅将 Amazon Personalize 搜索排名插件应用于开源 OpenSearch 中的查询。
6. 运行查询并比较结果。

通过开源 OpenSearch 监控插件

将 Amazon Personalize 搜索排名插件应用于 OpenSearch 查询时，您可以通过获取搜索管道的指标，监控插件。管道指标包括诸如 `personalized_search_ranking` 响应处理器的失败请求数量之类的统计信息。

您可以使用以下代码获取所有管道的指标。响应包含所有搜索管道的统计信息。有关管道指标的示例，请参阅管道指标示例。

```
curl -XGET "https://localhost:9200/_nodes/stats/search_pipeline?pretty" -ku
 'admin:admin'
```

`personalized_search_ranking` 响应处理器的字段

为 Amazon Personalize 搜索排名插件创建搜索管道时，指定具有以下字段的 `personalized_search_ranking` 响应处理器。

- campaign_arn（必填）- 指定用于对结果进行个性化设置的 Amazon Personalize 市场活动的 Amazon 资源名称 (ARN)。
- item_id_field（可选）- 如果 OpenSearch 中已编入索引的文档的 `_id` 字段与 Amazon Personalize itemId 不对应，请指定与之对应的字段的名称。默认情况下，该插件假定 `_id` 数据与 Amazon Personalize 数据中的 itemId 相匹配。
- 食谱（必填）- 指定要使用的 Amazon Personalize 食谱的名称。要使用该插件，您可以指定 `aws-personalized-ranking-v2` 或 `aws-personalized-ranking`。
- 权重（必填）- 指定响应处理器在重新对结果排名时对个性化的重视程度。指定一个介于 0.0 至 1.0 范围内的值。该值越接近 `1.0`，Amazon Personalize 结果排名越高的可能性越大。如果您指定 `0.0`，则不会进行个性化设置，且 OpenSearch 优先。
- 标签（可选）- 指定处理器的标识符。

- iam_role_arn（对 OpenSearch Service 为必填，对开源 OpenSearch 为可选）- 对于 OpenSearch Service，提供您在为 OpenSearch Service 设置权限以访问 Amazon Personalize 资源时创建的角色的 Amazon 资源名称（ARN）。如果您的 OpenSearch Service 和 Amazon Personalize 资源存在于不同的账户中，请指定为 OpenSearch Service 授予 `AssumeRole` 权限的角色。有关更多信息，请参阅 当资源位于不同账户时配置权限。

  对于开源 OpenSearch，如果您使用多个角色来限制组织中不同用户组的权限，请指定有权访问 Amazon Personalize 的角色的 ARN。如果您仅使用 OpenSearch 密钥库中的 AWS 凭证，则可以省略此字段。
- aws_region（必填）- 您创建 Amazon Personalize 市场活动时所在的 AWS 区域。
- ignore_failure（可选）- 指定插件是否忽略任何处理器故障。对于值，指定 `true` 或 `false`。对于您的生产环境，我们建议您指定 `true` 以免查询响应出现中断。对于测试环境，您可以指定 `false` 以查看插件生成的任何错误。
- external_account_iam_role_arn – 如果您使用 OpenSearch Service，并且您的 Amazon Personalize 和 OpenSearch Service 资源存在于不同的账户中，请指定有权访问 Amazon Personalize 资源的角色的 ARN。此角色必须与 Amazon Personalize 资源位于相同的账户中。有关更多信息，请参阅 当资源位于不同账户时配置权限。

有关 OpenSearch Service 代码示例，请参阅在 Amazon OpenSearch Service 中创建管道。有关开源 OpenSearch 示例，请参阅 在开源 OpenSearch 中创建管道。

管道指标示例

将 Amazon Personalize 搜索排名插件应用于 OpenSearch 查询时，您可以通过获取搜索管道的指标，监控插件。管道指标包括诸如 `personalized_search_ranking` 响应处理器的失败请求数量之类的统计信息。

以下代码显示了从 OpenSearch 返回的管道指标的摘录。它仅显示包含两个不同管道统计信息的 `pipelines` 对象。对于每个管道，您可以在 `personalized_search_ranking` 响应处理器列表中找到 Amazon Personalize 搜索排名插件指标。有关所有指标的完整示例，请参阅搜索管道指标。

```
{
....
....
  "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>
            }
          }
        }
      ]
    }
  }
....
....
}
```

为 Amazon Personalize 资源添加标签

标签是您可以选择定义的标签，并将其与 AWS 资源（包括某些类型的 Amazon Personalize 资源）相关联。一个资源最多可以有 50 个标签。

标签可帮助您以不同方式（例如按用途、环境或其他标准）对资源进行分类和管理。例如，您可以使用标签将收入分配给不同的职能，或者为不同的资源确定开发环境。

要按标签检索 Amazon 个性化资源，您可以使用 Resource Groups Tagging API 中的 GetResources 筛选条件。有关更多信息，请参阅 Res ource GetResourcesGroups Tagging API API 参考指南。

您可以向以下类型的 Amazon Personalize 资源添加标签：

- 批量推理作业
- 批量细分作业
- 市场活动
- 数据集
- 数据集组
- 数据集导入和导出作业
- 事件跟踪器
- 筛选条件
- 推荐器
- 解决方案
- 解决方案版本

主题

- 指南和要求
- 向 Amazon Personalize 资源添加标签
- 从 Amazon Personalize 资源中删除标签
- 在 IAM 策略中使用标签

指南和要求

每个标签都包含您定义的一个标签键和一个可选的标签值。标签键 是一种常见的标签，充当更具体的标签值的类别。标签值 充当标签键的描述符。

例如，如果您有两个版本的 Amazon Personalize 数据集组（一个用于内部测试，另一个用于生产），则您可为这两个项目分配一个 `Environment` 标签键。`Environment` 标签的标签值，对于其中一个版本的数据集组可以为 `Test`，对于另一个版本可以为 `Production`。

以下限制适用于标签：

- 每个资源的标签数上限：50
- 最大键长度：128 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度：256 个 Unicode 字符 (采用 UTF-8 格式)
- 标签键和值可以包含以下字符：A-Z、a-z、0-9、空格和 _ . : / = + @ –（连字符）。这是支持标签的 AWS 服务中可用的标准字符集。一些服务支持其他符号。
- 标签键和标签值区分大小写。
- 对于每个关联的资源，每个标签键都必须是唯一的，并且只能有一个标签值。
- 您的标签键和标签值不能以开头`aws:`。 AWS 服务应用以开头的标签`aws:`，并且这些标签无法修改。它们不计入标签限制。
- 您无法仅根据其标签更新或删除资源。您还必须指定 Amazon 资源名称 (ARN) 或资源 ID，具体取决于您使用的操作。

附加信息

有关标签的更多信息，请参阅以下资源。

- AWS AWS 一般参考文@@ 献中的标签原则
- AWS 标记策略（可下载的 PDF）
- AWS AWS IAM 用户指南中的@@ 访问控制
- AWS 《AWS Organizations 用户指南》中的@@ 标签策略

向 Amazon Personalize 资源添加标签

您可以使用 Amazon Personalize 控制台 AWS Command Line Interface （ AWS CLI ）或，在 Amazon Personalize 资源中添加、显示、更新和删除标签密钥和值 AWS SDKs。以下示例演示如何向 Amazon Personalize 数据集组添加标签。您可以用同样的方式向其他 Amazon Personalize 资源添加标签。

主题

- 添加标签（控制台）
- 添加标签 (AWS CLI)
- 添加标签 (AWS SDKs)

添加标签（控制台）

在 Amazon Personalize 中创建资源时，您可以使用 Amazon Personalize 控制台添加可选标签。以下示例向数据集组添加标签。

向新数据集组添加标签

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 选择创建数据集组。
3. 对于名称，输入名称。
4. 对于域，选择一个域。
5. 展开标签部分，选择添加新标签。
6. 对于键和值，输入对应的值。

   例如，分别为 **`Environment`** 和 **`Test`**。
7. 要添加更多标签，请选择添加新标签。

   您最多可以为每个资源添加 50 个标签。
8. 选择下一步来继续创建您的资源。

向现有资源添加标签的方法类似：选择您的资源并使用标签字段添加标签。

添加标签 (AWS CLI)

在创建资源或向现有资源添加标签时，可以使用 AWS Command Line Interface (AWS CLI) 来添加标签。

主题

- 在创建资源时添加标签
- 向现有资源添加标签

在创建资源时添加标签

要创建新资源并使用为其添加标签 AWS CLI，请对该资源使用相应的`create`命令并包括`tags`参数和值。例如，以下命令针对 ECOMMERCE 域创建一个名为 `myDatasetGroup` 的新域数据集组，并添加以下标签：带有 `Test` 标签值的 `Environment` 标签键以及 `Owner` 标签键和 `xyzCorp` 值。

```
aws personalize create-dataset-group \
--name myDatasetGroup \
--domain ECOMMERCE \
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

有关可用于创建 Amazon Personalize 资源的命令的信息，请参阅《亚马逊个性化 AWS CLI 命令参考》。

向现有资源添加标签

要向现有资源添加标签，请使用 `tag-resource` 命令。指定资源的 ARN，并在 `tags` 参数中提供标签键和值。

```
aws personalize tag-resource \
--resource-arn resource ARN \
--tags tagKey=key,tagValue=value
```

添加标签 (AWS SDKs)

您可以在创建资源时使用添加标签，或者为现有资源添加标签。 AWS SDKs

主题

- 在创建资源时添加标签

- 向现有资源添加标签

在创建资源时添加标签

要创建新资源并使用为其添加标签 AWS SDKs，请使用适当的`create`方法。使用 `tags` 参数为每个标签指定键值对。例如，以下代码针对 ECOMMERCE 域创建一个名为 `myDatasetGroup` 的新域数据集组，并添加以下标签：带有 `Test` 标签值的 `Environment` 标签键以及 `Owner` 标签键和 `xyzCorp` 值。

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 "";
}
```

向现有资源添加标签

以下代码展示了如何向现有 Amazon Personalize 资源添加标签。指定要为其添加标签的资源的 Amazon 资源名称 (ARN)，并为每个标签指定键值对。

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 "";
    }
```

从 Amazon Personalize 资源中删除标签

您可以使用 Amazon Personalize 控制台或使用 AWS Command Line Interface (AWS CLI) 或 UntagResource API 操作从 Amazon Personalize 资源中移除标签 AWS SDKs。以下示例演示如何从 Amazon Personalize 数据集组删除标签。您可以用同样的方式从其他 Amazon Personalize 资源中删除标签。

主题

- 删除标签（控制台）
- 删除标签（AWS CLI）
- 删除标签（AWS SDKs）

删除标签（控制台）

在 Amazon Personalize 中为资源添加标签后，您可以使用 Amazon Personalize 控制台删除标签。以下示例从数据集组中删除标签

从数据集组中删除标签

1. 在https://console.aws.amazon.com/personalize/家中打开 Amazon Personalize 控制台并登录您的账户。
2. 选择您的数据集组。
3. 在页面底部的标签选项卡中，选择管理标签。
4. 对于要删除的每个标签，选择删除。
5. 选择保存以删除标签。

删除标签（AWS CLI）

要使用从现有资源中移除标签 AWS CLI，请使用以下`untag-resource`命令。对于 `resource-arn`，指定源的 Amazon 资源名称（ARN）。对于 `tag-keys`，指定要删除的标签的键。

```
aws personalize untag-resource \
--resource-arn resource ARN \
--tag-keys key1 key2
```

删除标签（AWS SDKs）

要使用从现有 Amazon Personalize 资源中移除标签 AWS SDKs，请使用 UntagResource API 操作。以下代码演示如何使用 SDK for Python（Boto3）从数据集组中删除多个标签。对于 `resourceArn`，指定源的 Amazon 资源名称（ARN）。对于 `tagKeys`，指定要删除的标签的键。

```
import boto3

personalize = boto3.client('personalize')

response = personalize.untag_resource(
    resourceArn="Resource ARN",
    tagKeys=["tag1Key", "tag2Key"]
)
```

在 IAM 策略中使用标签

开始实施标签后，您可以向 AWS Identity and Access Management (IAM) 策略和 API 操作应用基于标记的资源级权限。这包括支持在创建资源时为资源添加标签的操作。通过以这种方式使用标签，您可以精细控制 AWS 账户中的哪些群组和用户有权创建和标记资源，以及哪些群组和用户有权更广泛地创建、更新和删除标签。

例如，您可以创建一个策略，允许用户只要其名称是 Amazon Personalize 资源的 `Owner` 标签中的值，就可以完全访问这些资源。

JSON

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

```
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

以下示例演示如何创建允许创建和删除数据集的策略。只有在用户名为 johndoe 时，才允许执行这些操作。

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": "*"
        }
    ]
}
```

如果您定义基于标签的资源级权限，该权限立即生效。这意味着，您的资源在创建后会更安全，而且您可以快速地开始将标签用于新资源。您还可以使用资源级权限来控制哪些标签键和值可以与新的和现有资源关联。有关更多信息，请参阅AWS 《IAM 用户指南》中的使用标签控制访问。

Amazon Personalize 常见问题解答

以下是与 Amazon Personalize 中导入数据、训练、模型部署、建议和筛选器相关的常见问题解答。

有关更多问题和答案，请参阅 Amazon Personalize 示例存储库中的 Amazon Personalize 备忘单。

主题

- 数据导入和管理
- 创建自定义解决方案和解决方案版本
- 模型部署（自定义市场活动）
- 建议
- 筛选建议

数据导入和管理

我的批量数据应该采用什么格式？

您的批量数据必须采用逗号分隔值 (CSV) 格式。CSV 文件的第一行必须包含列标题。CSV 文件中的列标题需要映射到此架构以创建数据集。如果您的数据包含任何非 ASCII 编码字符，则必须采用 UTF-8 格式编码 CSV 文件。请勿用引号 (") 将标题括起来。TIMESTAMP 和 CREATION_TIMESTAMP 数据必须采用 UNIX 纪元 时间格式。有关时间戳数据的更多信息，请参阅时间戳数据。有关 schema 的更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

有关完整的数据格式指南，请参阅为 Amazon Personalize 准备训练数据。如果您不确定如何格式化数据，可以使用 Amazon A SageMaker I Data Wrangler（Data Wrangler）来准备数据。有关更多信息，请参阅 使用 Amazon A SageMaker I Data Wrangler 准备和导入批量数据。

需要多少训练数据？

对于所有使用案例（域数据集组）和自定义食谱，您的交互数据必须具有以下内容：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个独立用户 IDs ，每个用户至少有两次物品互动。

为获得高质量的建议，我们建议您至少有 1000 名用户的最少 5 万次物品交互，每位用户有两次或更多次物品交互。

您可以从空的物品交互数据集开始，然后在记录了足够的数据后，仅使用新记录的事件创建推荐器（域数据集组）或自定义解决方案版本。某些食谱和使用案例可能有额外的数据要求。有关使用案例要求的信息，请参阅选择使用案例。有关食谱要求的信息，请参阅选择食谱。

如何更新物品或用户的属性？

使用 Amazon Personalize 控制台或 PutItems 或 PutUsers 操作，导入具有相同物品 ID 但属性已修改的物品或用户。

如何删除物品或用户？

Amazon Personalize 不支持删除特定物品或用户。要确保物品或用户不显示在建议中，请使用筛选器排除物品。有关更多信息，请参阅 筛选建议和用户细分。

如何删除架构？

您只能通过 DeleteSchema 操作删除架构。不能使用 Amazon Personalize 控制台删除架构。

创建自定义解决方案和解决方案版本

应使用何种食谱？

您使用的 Amazon Personalize 食谱因您的使用案例而定。有关将使用案例与食谱匹配的信息，请参阅选择食谱。Amazon Personalize 备忘单还包括使用案例和食谱信息。

我应该多久训练一次？

我们建议使用至少每周训练一次的自动训练频率。使用自动训练后，您就可以更轻松地维护推荐相关性。训练频率取决于业务需求、所使用的配方以及导入数据的频率。有关更多信息，请参阅 配置自动训练。有关维护相关性的信息，请参阅维护建议相关性。

是否应该使用 AutoML？

不应该使用。相反，我们建议您将使用案例与不同的 Amazon Personalize 食谱进行匹配，然后选择食谱。有关将使用案例与食谱匹配的信息，请参阅选择食谱。

模型部署（自定义市场活动）

我应该为我的市场活动的 minProvisionedTPS 设置什么？

高 minProvisionedTPS 会增加您的成本。我们建议从 1 开始设置 minProvisionedTPS（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要minProvisionedTPS时增加使用量。

如何监控我的市场活动费用？

Amazon Personalize Monitor 项目为 Amazon Personalize 活动提供了 CloudWatch 控制面板、自定义指标、利用率警报和成本优化功能。请参见 Amazon Personalize 示例存储库中的 Amazon Personalize Monitor。

如何为市场活动设置最大事务吞吐量？

您只能为市场活动设置最低 吞吐量。创建 Amazon Personalize 市场活动时，需要指定专用事务容量，用于为您的应用程序用户创建实时建议。如果您的 TPS 增加到超过 minProvisionedTPS，则 Amazon Personalize 会自动纵向扩展和缩减预调配容量，但永远不会低于 minProvisionedTPS。有关更多信息，请参阅 预调配的最低每秒事务数和自动扩缩。

建议

如何判断 Amazon Personalize 模型是否在生成质量建议？

使用离线和在线指标（参见通过指标评估 Amazon Personalize 解决方案版本）以及在线测试（例如A/B testing). For more information about A/B测试，请参阅）来评估解决方案版本的性能通过 A/B 测试衡量推荐的影响。

如何删除批量推理作业？为什么其状态为“活动”？

您无法删除批量推理作业。当批量推理作业的状态为活动时，表示作业已完成。您可以在输出 Amazon S3 存储桶或文件夹中访问您的建议。批量推理作业完成后，不会因此类作业而产生额外费用。但是，其他服务（例如 Amazon S3）可能会针对输入和输出数据存储收取额外费用。

为什么支持 SIMS 的市场活动会根据元数据推荐不相似的物品？

SIMS 使用您的物品交互数据集来确定相似度，而不是颜色或价格等物品元数据。SIMS 会识别交互数据集用户历史记录中物品的共存情况，以推荐相似物品。有关更多信息，请参阅 SIMS 配方。

我能否通过单个 GetRecommendations API 操作获得超过 500 个项目？

500 是您可以通过单个 GetRecommendations 检索的最大物品数。此值无法增加。

筛选建议

为什么没有按预期筛选建议？

有多种原因可能导致此错误：

- 筛选器表达式的格式或语法可能存在问题。有关格式正确的筛选器表达式的示例，请参阅筛选器表达式示例。
- Amazon Personalize 考虑每位用户每种事件类型最多 100 次最新交互。这是一个可调节的限额。您可以使用服务配额控制台请求增加配额。如果您有三个月没有导入某个用户的物品交互，则您的筛选器将不再考虑该用户的历史数据。要考虑这些数据，必须再次导入该用户的整个事件历史记录。

有关更多信息，请参阅 筛选建议和用户细分。

如何从建议中删除已购买的物品？

对于 ECOMMERCE 域数据集组，如果您为为您推荐或浏览 X 的客户也浏览过使用案例创建推荐器，则 Amazon Personalize 会根据您指定的 userId 和 `Purchase` 事件自动筛选用户购买的物品。

对于其他域数据集组使用案例或自定义资源，使用筛选器删除已购买的物品。向数据添加 `Purchased` 事件类型属性，通过 `PutItems` 操作记录购买 事件，并创建一个筛选器，从建议中删除已购买的物品。例如：

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("purchased")
```

有关更多信息，请参阅 筛选建议和用户细分。

Amazon Personalize 中的常见错误消息

以下部分列出和解释了您在使用 Amazon Personalize 时可能遇到的一些消息。

主题

- 数据导入和管理
- 创建解决方案和解决方案版本（自定义资源）
- 模型部署（自定义市场活动）
- 推荐器（域数据集组）
- 建议
- 筛选建议

数据导入和管理

错误消息：数据位置无效。

确保为您的 Amazon S3 存储桶位置使用了正确的语法。对于数据集导入作业，使用以下语法确定数据在 Amazon S3 中的位置：

s3://amzn-s3-demo-bucket/<folder path>/<CSVfilename>

如果您的 CSV 文件位于一个文件夹中，并且您想通过一个数据集导入作业上传多个文件，请使用不带 CSV 文件名的语法。

错误消息：调用 CreateDatasetImportJob 操作时出现错误 (LimitExceededException)：超过 5 个资源处于 PENDING 或 IN_PROGRESS 状态。

每个区域总共可以有 5 个待处理或正在进行中的数据集导入作业。此限额不可调整。有关 Amazon Personalize 配额的完整列表，请参阅 Amazon Personalize 端点和限额。

错误消息：Failed to create a data import job for <dataset type> dataset....Insufficient privileges for accessing data in Amazon S3.

通过将访问策略附加到您的 Amazon S3 存储桶和您的 Amazon Personalize 服务角色，向 Amazon Personalize 授予访问您 Amazon S3 资源的权限。请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

如果您使用 AWS Key Management Service (AWS KMS) 进行加密，则必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您的密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

错误消息：Failed to create a data import job <dataset type> dataset...Input CSV is missing the following columns:[COLUMN_NAME, COLUMN_NAME].

您导入 Amazon Personalize 的数据（包括属性名称和数据类型）必须与目标数据集的架构相匹配。有关更多信息，请参阅 为 Amazon Personalize 架构创建架构 JSON 文件。

错误消息：Length cannot be more than <character limit> characters for <COLLUMN_NAME>. 如果没有值超过字符限制，请确保您的数据符合 https://docs.aws.amazon.com/personalize/ latest/dg/data-prep-formatting.html 中列出的格式准则。

检查确保此列中的所有值都不超过字符限制。如果没有值超过字符限制，请检查前面的所有文本字段以确保以下内容：

- 确保所有文本数据都用双引号括起来。使用 \ 字符，对数据中的任何双引号或 \ 字符进行转义处理。
- 确保 CSV 文件中的每条记录都在单行中。

创建解决方案和解决方案版本（自定义资源）

错误消息：Create failed. Dataset has fewer than 25 users with at least 2 interactions each.

必须先导入更多数据，然后才能训练模型。训练模型的最低数据要求为：

- 至少 1000 条来自用户与您目录中的物品进行交互的物品交互记录。这些交互可以来自批量导入，也可以来自流事件，或者两者兼而有之。
- 至少 25 个独立用户 IDs ，每个用户至少有两次物品互动。

要获得实时建议，请使用数据集导入作业导入更多数据，或者使用事件跟踪器和 PutEvents 操作为用户记录更多交互事件。有关记录实时事件的更多信息，请参阅记录实时事件以影响推荐。

要获得批量建议，请在有更多数据时使用数据集导入作业导入数据。有关导入批量数据的更多信息，请参阅将训练数据导入 Amazon Personalize 数据集。

模型部署（自定义市场活动）

错误：Cannot create a campaign. More than 5 resources in ACTIVE state. Please delete some and try again.

每个数据集组总共可以有 5 个现用 Amazon Personalize 市场活动。此配额可调整，您可以使用服务限额控制台请求增加配额。有关 Amazon Personalize 限制和配额的完整列表，请参阅 Amazon Personalize 端点和限额。

推荐器（域数据集组）

错误：Dataset has fewer than 1000 interactions after filtering by event type: <event type>

不同的使用案例需要不同的事件类型。您的数据必须包含至少 1000 个事件，且其含有您的使用案例所需的类型。有关更多信息，请参阅 选择使用案例。

建议

批量推理作业错误消息：Invalid S3 input path or Invalid S3 output path

确保为您的 Amazon S3 输入或输出位置使用了正确的语法。此外，确保您的输出位置与输入数据不同。它应该是同一 Amazon S3 存储桶或不同存储桶中的一个文件夹。

在 Amazon S3 中对输入 文件位置使用以下语法：**s3://amzn-s3-demo-bucket/<folder name>/<input JSON file name>**

在 Amazon S3 中对输出 文件夹使用以下语法：**s3://amzn-s3-demo-bucket/<output folder name>/**

筛选建议

错误消息：Could not create filter. Invalid input symbol: $parameterName. Placeholders are not allowed with NOT_IN operator.

不能在使用 NOT_IN 运算符的筛选器表达式中使用占位符参数。相反，使用 IN 运算符并使用相反的 Action：使用 Include，而不是 Exclude（反之亦然）。

例如，如果要使用 `INCLUDE ItemID WHERE Items.GENRE NOT IN ($GENRE)`，则可以使用 `EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE)` 并获得相同的结果。

有关筛选条件的更多信息，请参阅筛选器表达式元素。

错误消息：Could not create filter. Invalid Expression... when filtering on Boolean type fields

您无法创建使用架构中布尔型值进行筛选的筛选器表达式。要根据布尔值进行筛选，请使用字段类型为 `String` 的架构，并在数据中使用值 `True` 和 `False`。或者，您可以使用类型 `int` 或 `long` 以及值 `0` 和 `1`。

有关筛选条件的更多信息，请参阅筛选器表达式元素。

使用指定资源 AWS CloudFormation

Amazon Personalize 与 AWS CloudFormation一项服务集成，可帮助您对 AWS 资源进行建模和设置，从而减少创建和管理资源和基础设施所花费的时间。您可以创建一个模板来描述您可以指定的所有 AWS 资源（例如 Amazon Personalize 数据集组）。 CloudFormation 然后为您预置和配置这些资源。

使用时 CloudFormation，您可以重复使用您的模板来一致且重复地设置您的 Amazon Personalize 资源。只需描述一次您的资源，然后在多个 AWS 账户 区域中一遍又一遍地配置相同的资源。

主题

- Amazon Persalize 和 CloudFormation 模板
- Amazon Personalize 资源的示例 CloudFormation 模板
- 了解更多关于 CloudFormation

Amazon Persalize 和 CloudFormation 模板

要为 Amazon Personalize 与相关服务预调配和配置资源，您必须了解 CloudFormation 模板。模板是 JSON 或 YAML 格式的文本文件。这些模板描述了您要在 CloudFormation 堆栈中配置的资源。如果你不熟悉 JSON 或 YAML，可以使用 D CloudFormation esigner 来帮助你开始使用 CloudFormation 模板。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的什么是 CloudFormation Designer？。

Amazon Personalize 支持在 CloudFormation中指定数据集、数据集组、数据集导入作业、架构和解决方案。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的 Amazon Personalize 资源类型参考。

Amazon Personalize 资源的示例 CloudFormation 模板

以下 CloudFormation 模板示例向您展示了如何指定不同的 Amazon Personalize 资源。

主题

- 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'
```

了解更多关于 CloudFormation

要了解更多信息 CloudFormation，请参阅以下资源：

- AWS CloudFormation
- AWS CloudFormation 用户指南
- CloudFormation API 参考
- AWS CloudFormation 命令行界面用户指南

使用 AWS SDK 的 Amazon Personalize 代码示例

以下代码示例演示了如何将 Amazon Personalize 与 AWS 软件开发工具包（SDK）结合使用。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

代码示例

- 使用 Amazon Personalize 的代码示例 AWS SDKs
  - 使用 Amazon Personalize 的基本示例 AWS SDKs
    - Amazon Personalize 的操作使用 AWS SDKs
      - 与 AWS SDK CreateBatchInferenceJob 配合使用
      - 与 AWS SDK CreateBatchSegmentJob 配合使用
      - 与 AWS SDK CreateCampaign 配合使用
      - 与 AWS SDK CreateDataset 配合使用
      - 与 AWS SDK CreateDatasetExportJob 配合使用
      - 与 AWS SDK CreateDatasetGroup 配合使用
      - 与 AWS SDK CreateDatasetImportJob 配合使用
      - 与 AWS SDK CreateEventTracker 配合使用
      - 与 AWS SDK CreateFilter 配合使用
      - 与 AWS SDK CreateRecommender 配合使用
      - 与 AWS SDK CreateSchema 配合使用
      - 与 AWS SDK CreateSolution 配合使用
      - 与 AWS SDK CreateSolutionVersion 配合使用
      - 与 AWS SDK DeleteCampaign 配合使用
      - 与 AWS SDK DeleteEventTracker 配合使用
      - 与 AWS SDK DeleteSolution 配合使用
      - 与 AWS SDK DescribeCampaign 配合使用
      - 与 AWS SDK DescribeRecipe 配合使用
      - 与 AWS SDK DescribeSolution 配合使用
      - 与 AWS SDK ListCampaigns 配合使用
      - 与 AWS SDK ListDatasetGroups 配合使用

            - 与 AWS SDK ListRecipes 配合使用
            - 与 AWS SDK ListSolutions 配合使用
            - 与 AWS SDK UpdateCampaign 配合使用
- 使用 Amazon Personalize 事件的代码示例 AWS SDKs
  - 使用 Amazon Personalize 事件的基本示例 AWS SDKs
    - 使用 Amazon Personalize 对事件进行操作 AWS SDKs
      - 与 AWS SDK PutEvents 配合使用
      - 与 AWS SDK PutItems 配合使用
      - 与 AWS SDK PutUsers 配合使用
- 使用 Amazon Personalize 运行时的代码示例 AWS SDKs
  - 使用 Amazon Personalize 运行时的基本示例 AWS SDKs
    - Amazon Personalize 运行时的操作使用于 AWS SDKs
      - 与 AWS SDK GetPersonalizedRanking 配合使用
      - 与 AWS SDK GetRecommendations 配合使用

使用 Amazon Personalize 的代码示例 AWS SDKs

以下代码示例展示了如何将 Amazon Personalize 与 AWS 软件开发套件 (SDK) 配合使用。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon Personalize

- 使用 Amazon Personalize 的基本示例 AWS SDKs
  - Amazon Personalize 的操作使用 AWS SDKs
    - 与 AWS SDK CreateBatchInferenceJob 配合使用
    - 与 AWS SDK CreateBatchSegmentJob 配合使用
    - 与 AWS SDK CreateCampaign 配合使用
    - 与 AWS SDK CreateDataset 配合使用
    - 与 AWS SDK CreateDatasetExportJob 配合使用

- 与 AWS SDK CreateDatasetGroup 配合使用
- 与 AWS SDK CreateDatasetImportJob 配合使用
- 与 AWS SDK CreateEventTracker 配合使用
- 与 AWS SDK CreateFilter 配合使用
- 与 AWS SDK CreateRecommender 配合使用
- 与 AWS SDK CreateSchema 配合使用
- 与 AWS SDK CreateSolution 配合使用
- 与 AWS SDK CreateSolutionVersion 配合使用
- 与 AWS SDK DeleteCampaign 配合使用
- 与 AWS SDK DeleteEventTracker 配合使用
- 与 AWS SDK DeleteSolution 配合使用
- 与 AWS SDK DescribeCampaign 配合使用
- 与 AWS SDK DescribeRecipe 配合使用
- 与 AWS SDK DescribeSolution 配合使用
- 与 AWS SDK ListCampaigns 配合使用
- 与 AWS SDK ListDatasetGroups 配合使用
- 与 AWS SDK ListRecipes 配合使用
- 与 AWS SDK ListSolutions 配合使用
- 与 AWS SDK UpdateCampaign 配合使用

使用 Amazon Personalize 的基本示例 AWS SDKs

以下代码示例展示了如何使用 Amazon Personalize 的基础知识 AWS SDKs。

示例

- Amazon Personalize 的操作使用 AWS SDKs
  - 与 AWS SDK CreateBatchInferenceJob 配合使用
  - 与 AWS SDK CreateBatchSegmentJob 配合使用
  - 与 AWS SDK CreateCampaign 配合使用
  - 与 AWS SDK CreateDataset 配合使用
  - 与 AWS SDK CreateDatasetExportJob 配合使用

- 与 AWS SDK CreateDatasetGroup 配合使用
- 与 AWS SDK CreateDatasetImportJob 配合使用
- 与 AWS SDK CreateEventTracker 配合使用
- 与 AWS SDK CreateFilter 配合使用
- 与 AWS SDK CreateRecommender 配合使用
- 与 AWS SDK CreateSchema 配合使用
- 与 AWS SDK CreateSolution 配合使用
- 与 AWS SDK CreateSolutionVersion 配合使用
- 与 AWS SDK DeleteCampaign 配合使用
- 与 AWS SDK DeleteEventTracker 配合使用
- 与 AWS SDK DeleteSolution 配合使用
- 与 AWS SDK DescribeCampaign 配合使用
- 与 AWS SDK DescribeRecipe 配合使用
- 与 AWS SDK DescribeSolution 配合使用
- 与 AWS SDK ListCampaigns 配合使用
- 与 AWS SDK ListDatasetGroups 配合使用
- 与 AWS SDK ListRecipes 配合使用
- 与 AWS SDK ListSolutions 配合使用
- 与 AWS SDK UpdateCampaign 配合使用

Amazon Personalize 的操作使用 AWS SDKs

以下代码示例演示了如何使用执行单独的 Amazon Personalize 操作 AWS SDKs。每个示例都包含一个指向的链接 GitHub，您可以在其中找到有关设置和运行代码的说明。

以下示例仅包括最常用的操作。有关完整列表，请参阅 Amazon Personalize API 参考。

示例

- 与 AWS SDK CreateBatchInferenceJob 配合使用
- 与 AWS SDK CreateBatchSegmentJob 配合使用
- 与 AWS SDK CreateCampaign 配合使用
- 与 AWS SDK CreateDataset 配合使用

- 与 AWS SDK CreateDatasetExportJob 配合使用
- 与 AWS SDK CreateDatasetGroup 配合使用
- 与 AWS SDK CreateDatasetImportJob 配合使用
- 与 AWS SDK CreateEventTracker 配合使用
- 与 AWS SDK CreateFilter 配合使用
- 与 AWS SDK CreateRecommender 配合使用
- 与 AWS SDK CreateSchema 配合使用
- 与 AWS SDK CreateSolution 配合使用
- 与 AWS SDK CreateSolutionVersion 配合使用
- 与 AWS SDK DeleteCampaign 配合使用
- 与 AWS SDK DeleteEventTracker 配合使用
- 与 AWS SDK DeleteSolution 配合使用
- 与 AWS SDK DescribeCampaign 配合使用
- 与 AWS SDK DescribeRecipe 配合使用
- 与 AWS SDK DescribeSolution 配合使用
- 与 AWS SDK ListCampaigns 配合使用
- 与 AWS SDK ListDatasetGroups 配合使用
- 与 AWS SDK ListRecipes 配合使用
- 与 AWS SDK ListSolutions 配合使用
- 与 AWS SDK UpdateCampaign 配合使用

与 AWS SDK **CreateBatchInferenceJob** 配合使用

以下代码示例演示如何使用 `CreateBatchInferenceJob`。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
       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 "";
        }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateBatchInferenceJob中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateBatchInferenceJob中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateBatchSegmentJob** 配合使用

以下代码示例演示了如何使用 CreateBatchSegmentJob。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateBatchSegmentJob中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateCampaign** 配合使用

以下代码示例演示如何使用 CreateCampaign。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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;
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateCampaign中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateCampaign中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateDataset** 配合使用

以下代码示例演示如何使用 CreateDataset。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateDataset中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateDataset中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateDatasetExportJob** 配合使用

以下代码示例演示如何使用 CreateDatasetExportJob。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateDatasetExportJob中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateDatasetExportJob中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateDatasetGroup** 配合使用

以下代码示例演示如何使用 CreateDatasetGroup。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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 "";
   }
```

创建域数据集组。

```
   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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateDatasetGroup中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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);
```

创建域数据集组。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateDatasetGroup中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateDatasetImportJob** 配合使用

以下代码示例演示如何使用 CreateDatasetImportJob。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考[CreateDatasetImportJob](#)中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateDatasetImportJob中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateEventTracker** 配合使用

以下代码示例演示如何使用 CreateEventTracker。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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;
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateEventTracker中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateEventTracker中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateFilter** 配合使用

以下代码示例演示如何使用 `CreateFilter`。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
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 "";
}
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateFilter中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateFilter中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateRecommender** 配合使用

以下代码示例演示如何使用 CreateRecommender。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateRecommender中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateRecommender中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateSchema** 配合使用

以下代码示例演示如何使用 CreateSchema。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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 "";
   }
```

创建含域的架构。

```
   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 "";
   }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考[CreateSchema](#)中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

创建含域的架构。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateSchema中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateSolution** 配合使用

以下代码示例演示如何使用 CreateSolution。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateSolution中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateSolution中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **CreateSolutionVersion** 配合使用

以下代码示例演示如何使用 CreateSolutionVersion。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考CreateSolutionVersion中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考CreateSolutionVersion中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DeleteCampaign** 配合使用

以下代码示例演示了如何使用 DeleteCampaign。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
}
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DeleteCampaign中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DeleteEventTracker** 配合使用

以下代码示例演示了如何使用 DeleteEventTracker。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DeleteEventTracker中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DeleteSolution** 配合使用

以下代码示例演示了如何使用 DeleteSolution。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DeleteSolution中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DescribeCampaign** 配合使用

以下代码示例演示了如何使用 DescribeCampaign。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DescribeCampaign中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DescribeRecipe** 配合使用

以下代码示例演示了如何使用 DescribeRecipe。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DescribeRecipe中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **DescribeSolution** 配合使用

以下代码示例演示了如何使用 DescribeSolution。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考DescribeSolution中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **ListCampaigns** 配合使用

以下代码示例演示了如何使用 ListCampaigns。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考ListCampaigns中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **ListDatasetGroups** 配合使用

以下代码示例演示了如何使用 ListDatasetGroups。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考ListDatasetGroups中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **ListRecipes** 配合使用

以下代码示例演示了如何使用 ListRecipes。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考ListRecipes中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **`ListSolutions`** 配合使用

以下代码示例演示了如何使用 `ListSolutions`。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考ListSolutions中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **UpdateCampaign** 配合使用

以下代码示例演示了如何使用 UpdateCampaign。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    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 "";
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考UpdateCampaign中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

使用 Amazon Personalize 事件的代码示例 AWS SDKs

以下代码示例展示了如何将 Amazon Personalize Even AWS ts 与软件开发套件 (SDK) 配合使用。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon Personalize Events

- 使用 Amazon Personalize 事件的基本示例 AWS SDKs
  - 使用 Amazon Personalize 对事件进行操作 AWS SDKs
    - 与 AWS SDK PutEvents 配合使用
    - 与 AWS SDK PutItems 配合使用
    - 与 AWS SDK PutUsers 配合使用

使用 Amazon Personalize 事件的基本示例 AWS SDKs

以下代码示例展示了如何使用 Amazon Personalize Events 的基础知识 AWS SDKs。

示例

- 使用 Amazon Personalize 对事件进行操作 AWS SDKs
  - 与 AWS SDK PutEvents 配合使用
  - 与 AWS SDK PutItems 配合使用
  - 与 AWS SDK PutUsers 配合使用

使用 Amazon Personalize 对事件进行操作 AWS SDKs

以下代码示例演示了如何使用执行单独的 Amazon Personalize Events 操作 AWS SDKs。每个示例都包含一个指向的链接 GitHub，您可以在其中找到有关设置和运行代码的说明。

以下示例仅包括最常用的操作。有关完整列表，请参阅 Amazon Personalize Events API 参考。

示例

- 与 AWS SDK PutEvents 配合使用
- 与 AWS SDK PutItems 配合使用
- 与 AWS SDK PutUsers 配合使用

与 AWS SDK **PutEvents** 配合使用

以下代码示例演示如何使用 PutEvents。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
       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;
        }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考PutEvents中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考PutEvents中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **PutItems** 配合使用

以下代码示例演示了如何使用 PutItems。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考PutItems中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **PutUsers** 配合使用

以下代码示例演示如何使用 PutUsers。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
       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;
        }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考PutUsers中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考PutUsers中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

使用 Amazon Personalize 运行时的代码示例 AWS SDKs

以下代码示例展示了如何将 Amazon Personalize Runtime 与 AWS 软件开发套件 (SDK) 配合使用。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon Personalize Runtime

- 使用 Amazon Personalize 运行时的基本示例 AWS SDKs
  - Amazon Personalize 运行时的操作使用于 AWS SDKs
    - 与 AWS SDK GetPersonalizedRanking 配合使用
    - 与 AWS SDK GetRecommendations 配合使用

使用 Amazon Personalize 运行时的基本示例 AWS SDKs

以下代码示例展示了如何使用 Amazon Personalize Runtime 的基础知识 AWS SDKs。

示例

- Amazon Personalize 运行时的操作使用于 AWS SDKs
  - 与 AWS SDK GetPersonalizedRanking 配合使用
  - 与 AWS SDK GetRecommendations 配合使用

Amazon Personalize 运行时的操作使用于 AWS SDKs

以下代码示例演示了如何使用执行单独的 Amazon Personalize 运行时操作 AWS SDKs。每个示例都包含一个指向的链接 GitHub，您可以在其中找到有关设置和运行代码的说明。

以下示例仅包括最常用的操作。有关完整列表，请参阅 Amazon Personalize Runtime API 参考。

示例

- 与 AWS SDK GetPersonalizedRanking 配合使用
- 与 AWS SDK GetRecommendations 配合使用

与 AWS SDK **GetPersonalizedRanking** 配合使用

以下代码示例演示如何使用 `GetPersonalizedRanking`。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   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;
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考GetPersonalizedRanking中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考GetPersonalizedRanking中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

与 AWS SDK **GetRecommendations** 配合使用

以下代码示例演示如何使用 GetRecommendations。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

获取推荐项目列表。

```
   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);
        }
    }
```

从在域数据集组中创建的推荐系统获取推荐项目列表。

```
    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);
        }
```

```
    }
```

请求推荐时使用筛选条件。

```
   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);
        }
    }
```

- 有关 API 的详细信息，请参阅 AWS SDK for Java 2.x API 参考GetRecommendations中的。

JavaScript

适用于 JavaScript (v3) 的软件开发工具包

> **Note**
>
> 还有更多相关信息 GitHub。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
// 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();
```

使用筛选条件（自定义数据集组）获取推荐。

```
// 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();
```

从在域数据集组中创建的推荐系统处获取经过筛选的推荐。

```
// 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();
```

- 有关 API 的详细信息，请参阅 适用于 JavaScript 的 AWS SDK API 参考GetRecommendations中的。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅将 Amazon Personalize 与 AWS SDK。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon Personalize 中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模型将此描述为云的安全性与云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。Amazon Personalize 使用数据加密来保护您的数据。有关更多信息，请参阅Amazon Personalize 中的数据加密。 Third-party作为AWS 合规计划合规计划合规计划合的一部分，审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon Personalize 的合规性计划，请参阅按合规性计划提供的范围内AWS 服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 Amazon Personalize 时应用责任共担模式。以下主题说明如何配置 Amazon Personalize 以实现您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 Amazon Personalize 资源。

主题

- Amazon Personalize 中的数据保护
- 适用于 Amazon Personalize 的身份和访问管理
- 监控亚马逊 Personalize 使用亚马逊进行个性化设置 CloudWatch
- 使用记录亚马逊个性化 API 调用 AWS CloudTrail
- Amazon Personalize 的合规性验证
- Amazon Personalize 中的故障恢复能力
- Amazon Personalize 中的基础设施安全性
- Amazon Personalize 和接口 VPC 端点 (AWS PrivateLink)

Amazon Personalize 中的数据保护

分担责任模式 AWS 分适用于 Amazon Personalize 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责

您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题解答AWS条款。 有关欧洲数据保护的信息，请参阅通用数据保护条例（GDPR）中心。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您 AWS 服务 使用控制台、API 或软件开发工具包与 Amazon Per AWS sonalize 或其他人合作时。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

Amazon Personalize 中的数据加密

以下信息说明 Amazon Personalize 在何处使用数据加密来保护您的数据。

静态加密

存储在 Amazon Personalize 中的任何数据始终使用亚马逊个性化托管 AWS Key Management Service (AWS KMS) 密钥进行静态加密。如果您在创建资源时提供了自己的 AWS KMS 密钥，则 Amazon Personalize 会使用该密钥来加密您的数据并将其存储。例如，如果您在 CreateDatasetGroup 操作中提供了 AWS KMS ARN，则 Amazon Personalize 将使用该密钥来加密并存储您导入到您在该数据集组中创建的任何数据集中的数据。

您必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

有关 Amazon S3 中数据加密的信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用加密保护数据。有关管理自己的 AWS KMS 密钥的信息，请参阅《AWS Key Management Service 开发者指南》中的管理密钥。

传输中加密

Amazon Personalize 使用带有 AWS 证书的 TLS 来加密发送到其他 AWS 服务的任何数据。与其他 AWS 服务的任何通信都通过 HTTPS 进行，而 Amazon Personalize 终端节点仅支持通过 HTTPS 进行安全连接。

Amazon Personalize 会将数据从您的账户中复制出来，并在内部 AWS 系统中进行处理。处理数据时，Amazon Personalize 会使用亚马逊个性化 AWS KMS 密钥或您提供的任何密 AWS KMS 钥对数据进行加密。

密钥管理

AWS 管理所有默认 AWS KMS 密钥。您有责任管理自己拥有的所有 AWS KMS 密钥。您必须向 Amazon Personalize 和 Amazon Personalize IAM 服务角色授予使用您密钥的权限。有关更多信息，请参阅 授予 Amazon Personalize 使用您的 AWS KMS 密钥的权限。

有关管理自己的 AWS KMS 密钥的信息，请参阅《AWS Key Management Service 开发者指南》中的管理密钥。

适用于 Amazon Personalize 的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（具有权限）来使用 Amazon Personalize 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- 如何配合使用 Amazon Personalize 与 IAM
- Cross-service 混乱的副手预防
- Identity-based Amazon Personalize 政策示例
- 排查 Amazon Personalize 身份和访问问题

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅排查 Amazon Personalize 身份和访问问题）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅如何配合使用 Amazon Personalize 与 IAM）
- IAM 管理员：编写用于管理访问权限的策略（请参阅Identity-based Amazon Personalize 政策示例）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

AWS 账户 根用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。 AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

Identity-based 政策

Identity-based 策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

Identity-based 策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

Resource-based 政策

Resource-based 策略是您附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中指定主体。

Resource-based 策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略（ SCP ）– 指定 AWS Organizations中组织或组织单元的最大权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略（ RCP ）– 设置对账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的资源控制策略（ RCP ）。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

如何配合使用 Amazon Personalize 与 IAM

在使用 IAM 管理对 Amazon Personalize 的访问权限之前，了解哪些 IAM 特征可用于 Amazon Personalize。

可与 Amazon Personalize 一起使用的 IAM 特征

IAM 功能	Amazon Personalize 支持
Identity-based 政策	是

IAM 功能	Amazon Personalize 支持
Resource-based 政策	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是
ACL	否
ABAC（策略中的标签）	是
临时凭证	是
主体权限	是
服务角色	是
Service-linked 角色	否

要全面了解 Amazon Personalize 和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中与 IAM 配合使用的AWS 服务。

Identity-based Amazon Personalize 政策

支持基于身份的策略：是

Identity-based 策略是您可以附加到身份（例如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Identity-based Amazon Personalize 政策示例

要查看 Amazon Personalize 基于身份的策略的示例，请参阅 Identity-based Amazon Personalize 政策示例。

Resource-based 亚马逊 Personalize 内部的政策

支持基于资源的策略：否

Resource-based 策略是您附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

Amazon Personalize 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 Amazon Personalize 操作的列表，请参阅《服务授权参考》中的 Amazon Personalize 定义的操作。

Amazon Personalize 中的策略操作在操作前面使用以下前缀：

```
personalize
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "personalize:action1",
      "personalize:action2"
         ]
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，包括以下操作：

```
"Action": "personalize:Describe*"
```

要查看 Amazon Personalize 基于身份的策略的示例，请参阅 Identity-based Amazon Personalize 政策示例。

Amazon Personalize 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 Amazon Personalize 资源类型及其 ARN 的列表，请参阅《服务授权参考》中的由 Amazon Personalize 定义的资源。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 Amazon Personalize 定义的操作。

要查看 Amazon Personalize 基于身份的策略的示例，请参阅 Identity-based Amazon Personalize 政策示例。

Amazon Personalize 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Condition` 元素根据定义的条件指定语句何时执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 Amazon Personalize 条件键的列表，请参阅《服务授权参考》中的 Amazon Personalize 的条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 Amazon Personalize 定义的操作。

要查看 Amazon Personalize 基于身份的策略的示例，请参阅 Identity-based Amazon Personalize 政策示例。

Amazon Personalize 中的 ACL

支持 ACL：否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，但它们不使用 JSON 策略文档格式。

Amazon Personalize 的 ABAC

支持 ABAC（策略中的标签）：是

Attribute-based 访问控制 (ABAC) 是一种授权策略，它根据称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/`*`key-name`*`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

有关标记 Amazon Personalize 资源的更多信息，请参阅为 Amazon Personalize 资源添加标签。

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅 在 IAM 策略中使用标签。

将临时凭证用于 Amazon Personalize

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

Cross-service Amazon Personalize 的主要权限

支持转发访问会话（FAS）：是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

Amazon Personalize 的服务角色

支持服务角色：是

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会破坏 Amazon Personalize 的功能。仅当 Amazon Personalize 提供相关指导时才编辑服务角色。

Service-linked 亚马逊 Personalize 的角色

支持服务相关角色：否

服务相关角色是一种链接到的服务角色。 AWS 服务该服务可以代替您执行操作。 Service-linked 角色出现在您的， AWS 账户 并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅能够与 IAM 搭配使用的AWS 服务。在表中查找Service-linked 角色列Yes中包含的服务。选择是链接以查看该服务的服务相关角色文档。

Cross-service 混乱的副手预防

混淆代理问题是一个安全性问题，即不具有某操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。 Cross-service 当一个服务（调用服务）调用另一个服务（被调用的服务）时，可能会发生模仿行为。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况， AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用 `aws:SourceArn` 或 `aws:SourceAccount` 全局条件上下文键，以限制 Amazon Personalize 为其他服务提供的资源访问权限。

为了防止在 Amazon Personalize 担任的角色中出现混淆代理问题，在该角色的信任策略中，将 `aws:SourceArn` 的值设置为 `arn:aws:personalize:region:accountNumber:*`。通配符 (*) 对所有 Amazon Personalize 资源应用条件。

以下信任关系策略向 Amazon Personalize 授予对资源的访问权限，并使用 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键来防范混淆代理问题。在为 Amazon Personalize 创建角色（为 Amazon Personalize 创建 IAM 角色）时，使用此策略。

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 Amazon Personalize 政策示例

默认情况下，用户和角色没有创建或修改 Amazon Personalize 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 Amazon Personalize 定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》中的 Amazon Personalize 的操作、资源和条件键。

主题

- 策略最佳实践
- AWS 托管策略
- 使用 Amazon Personalize 控制台
- 允许用户查看他们自己的权限
- 允许完全访问 Amazon Personalize 资源
- 允许对 Amazon Personalize 资源的只读访问

策略最佳实践

Identity-based 政策决定了是否有人可以在您的账户中创建、访问或删除亚马逊个性化资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管策略或工作职能的AWS 托管策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

AWS 托管策略

AWS 托管策略是由 AWS创建和管理的策略。以下是您在使用 Amazon Personalize 时可能使用的 AWS 托管政策示例。

AmazonPersonalizeFullAccess 策略

您可以使用 AWS 托管`AmazonPersonalizeFullAccess`策略向用户授予以下权限：

- 访问所有 Amazon Personalize 资源
- 在 Amazon 上发布和列出指标 CloudWatch
- 列出、读取、写入和删除 Amazon S3 存储桶中将 `Personalize` 或 `personalize` 包含在存储桶名称中的所有对象
- 将角色传递给 Amazon Personalize

`AmazonPersonalizeFullAccess` 提供的权限超出了必要的范围。我们建议创建一个仅授予必要权限的新 IAM 策略（请参阅向 Amazon Personalize 授予访问资源的权限）。

CloudWatchFullAccess

要向您的用户授予监控 Amazon Personalize 的权限 CloudWatch，请将该`CloudWatchFullAccess`政策附加到您的角色中。有关更多信息，请参阅 监控亚马逊 Personalize 使用亚马逊进行个性化设置 CloudWatch。

`CloudWatchFullAccess` 策略为可选策略，可授予执行以下操作的权限：

- 在中发布和列出 Amazon Personalize 指标 CloudWatch
- 查看指标和指标统计信息。
- 设置基于指标的警报。

使用 Amazon Personalize 控制台

要访问 Amazon Personalize 控制台，您必须具有一组最低的权限。这些权限必须允许您列出和查看有关您的 Amazon Personalize 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "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": "*"
        }
    ]
}
```

允许完全访问 Amazon Personalize 资源

以下示例允许您 AWS 账户中的 IAM 用户完全访问所有 Amazon Personalize 资源和操作。

JSON

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

允许对 Amazon Personalize 资源的只读访问

在此示例中，您向 AWS 账户中的 IAM 用户授予对您的 Amazon Personalize 资源的只读访问权限，包括亚马逊个性化数据集、数据集组、解决方案和活动。

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": "*"
        }
    ]
}
```

排查 Amazon Personalize 身份和访问问题

使用以下信息，帮助您诊断和修复在使用 Amazon Personalize 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 Amazon Personalize 中执行操作
- 我无权执行 iam：PassRole
- 我想允许我以外的人进入 AWS 账户 访问我的 Amazon Personalize 资源

我无权在 Amazon Personalize 中执行操作

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 personalize:*GetWidget* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 personalize:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 personalize:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，表明您无权执行 iam:PassRole 操作，则必须更新策略以允许您将角色传递给 Amazon Personalize。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 Amazon Personalize 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人进入 AWS 账户 访问我的 Amazon Personalize 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon Personalize 是否支持这些特征，请参阅如何配合使用 Amazon Personalize 与 IAM。

- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

监控亚马逊 Personalize 使用亚马逊进行个性化设置 CloudWatch

通过亚马逊 CloudWatch，您可以获得与 Amazon Personalize 相关的指标。您可以设置警报，以便在一个或多个指标超出定义的阈值时通知您。要查看指标，您可以使用亚马逊 CloudWatch AWS Command Line Interface、亚马逊或 CloudWatch API。

主题

- 使用 Amazon Personalize 的 CloudWatch 指标
- 访问 Amazon Personalize 指标
- 创建警报
- Amazon Personalize 无服务器监控应用程序示例
- CloudWatch 亚马逊 Personalize 的指标

使用 Amazon Personalize 的 CloudWatch 指标

要使用指标，您必须指定以下信息：

- 指标名称。
- 指标维度。维度 是帮助您对某指标进行唯一标识的名称/值对。

您可以使用 AWS 管理控制台、或 CloudWatch API 获取 Amazon Personal AWS CLI ize 的监控数据。您也可以通过其中一个 AWS 软件开发工具包或 CloudWatch AP CloudWatch I 工具来使用 API。控制台根据来自 CloudWatch API 的原始数据显示一系列图表。根据您的需求差异，您可能倾向于使用控制台中显示的图表，也可能倾向于检索自 API 的图表。

下面的列表显示这些指标的一些常见用途。这些是入门建议，并不全面。

如何？	相关指标
如何跟踪已记录的事件数。	监控 PutEventsRequests 指标。
如何监控 DatasetImportJob 错误？	使用 DatasetImportJobError 指标。
我如何监控 GetRecommendations 调用的延迟？	使用 GetRecommendationsLatency 指标。

您必须拥有相应的 CloudWatch 权限才能使用监控 Amazon Personalize CloudWatch。有关更多信息，请参阅 Amazon 的身份验证和访问控制 CloudWatch。

访问 Amazon Personalize 指标

以下示例说明如何使用 CloudWatch 控制台、和 CloudWatch API 访问 Amazon Person AWS CLI alize 指标。

要查看指标（控制台）

1. 登录 AWS 管理控制台 并打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 选择指标，选择所有指标选项卡，然后选择 AWS/Personalize。
3. 选择指标维度。
4. 从列表中选择所需的指标，然后为图表选择时间段。

查看在一段时间内收到的事件的指标 (CLI)

- 打开 AWS CLI 并输入以下命令：

```
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
```

此示例显示在一段时间内为给定事件跟踪程序 ARN 接收的事件。有关更多信息，请参阅 get-metric-statistics。

访问指标 (CloudWatch API)

- 调用 GetMetricStatistics。有关更多信息，请参阅 Amazon CloudWatch API 参考。

创建警报

您可以创建一个 CloudWatch 警报，在警报状态发生变化时发送亚马逊简单通知服务 (Amazon SNS) Simple Notification Service 消息。警报会在您规定的时间范围内监控某一项指标。警报会根据指标值在多个时间段内对比给定阈值的情况执行一项或多项操作。操作是一个发送到 Amazon SNS 主题或 AWS Auto Scaling 策略的通知。

警报仅针对持续的状态变化调用操作。 CloudWatch 警报不会仅仅因为它们处于特定状态就调用操作。该状态必须改变并在指定数量的时间段内一直保持。

设置警报（控制台）

1. 登录 AWS 管理控制台 并打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择警报，然后选择创建警报。创建警报向导随即启动。
3. 选择选择指标。
4. 在所有指标选项卡中，选择 AWS/Personalize。
5. 选择 EventTrackerArn，然后选择PutEventsRequests指标。
6. 选择绘成图表的指标选项卡。
7. 对于统计数据，选择总计。
8. 选择选择指标。
9. 填写名称和描述。对于每当，选择 >，然后输入您选择的最大值。
10. 如果 CloudWatch 要在达到警报状态时向您发送电子邮件，请在 “每当此警报:” 中选择 “状态为警报”。要将警报发送到现有 Amazon SNS 主题，对于发送通知到：，请选择现有 SNS 主题。要为新的电子邮件订阅列表设置名称和电子邮件地址，请选择 “新建列表”。 CloudWatch 保存列表并将其显示在字段中，这样您就可以用它来设置 future 的警报。

> **Note**
>
> 如果您使用新列表创建一个新的 Amazon SNS 主题，则必须先验证电子邮件地址，然后目标收件人才能接收通知。Amazon SNS 仅在警报进入警报状态时发送电子邮件。如果在验证电子邮件地址之前警报状态发生了变化，那么目标收件人不会接收到通知。

11. 选择创建警报。

要设置闹钟 (AWS CLI)

- 打开 AWS CLI，然后输入以下命令。更改 `alarm-actions` 参数的值，以引用您之前创建的 Amazon SNS 主题。

```
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
```

此示例显示如何为在 5 分钟内针对给定事件跟踪程序 ARN 发生 10 个以上的事件的情况创建警报。有关更多信息，请参阅 put-metric-alarm。

设置警报 (CloudWatch API)

- 调用 `PutMetricAlarm`。有关更多信息，请参阅 Amazon CloudWatch API 参考。

Amazon Personalize 无服务器监控应用程序示例

有关为 Amazon Personalize 添加监控、警报和优化功能的示例应用程序，请参阅 Amazon Personalize 示例存储库中的 Amazon Personalize Monitor。

CloudWatch 亚马逊 Personalize 的指标

本部分包含有关可用于 Amazon Personalize 的亚马逊 CloudWatch 指标的信息。有关更多信息，请参阅 监控亚马逊 Personalize 使用亚马逊进行个性化设置 CloudWatch。

下表列出了 Amazon Personalize 指标。除 GetRecommendations 并 GetPersonalizedRanking 支持这些统计数据之外的所有指标:`Average`, `Minimum`, `Maximum`, `Sum`. GetRecommendations 并且Sum仅 GetPersonalizedRanking 支持。

指标	说明
`DatasetImportJobRequests`	成功的 CreateDatasetImportJob API 调用的数量。 维度：`DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn`
`DatasetImportJobError`	导致错误的 `CreateDatasetImportJob` API 调用的数量。 维度：`DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn`
`DatasetImportJobExecutionTime`	`CreateDatasetImportJob` API 调用与操作完成（或发生故障）之间的时间。 维度：`DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn` 单位：秒
`DatasetSize`	由数据集导入任务导入的数据的大小。 维度：`DatasetGroupArn`, `DatasetArn`, `DatasetImportJobArn` 单位：字节
`SolutionTrainingJobRequests`	成功的 CreateSolutionVersion API 调用的数量。 维度：`SolutionArn`, `SolutionVersionArn`

指标	说明
SolutionTrainingJobError	导致错误的 CreateSolutionVersion API 调用的数量。 维度：SolutionArn, SolutionVersionArn
SolutionTrainingJobExecutionTime	CreateSolutionVersion API 调用与操作完成（或发生故障）之间的时间。 维度：SolutionArn, SolutionVersionArn 单位：秒
GetPersonalizedRanking	GetPersonalizedRanking API 调用是否成功。使用sum统计数据查看成功 GetPersonalizedRanking API 调用的总数。此指标不支持其他统计信息。 尺寸：CampaignArn
GetPersonalizedRanking4xxErrors	4xx HTTP 响应代码返回的 GetPersonalizedRanking API 调用的数量。 尺寸：CampaignArn
GetPersonalizedRanking5xxErrors	5xx HTTP 响应代码返回的 GetPersonalizedRanking API 调用的数量。 尺寸：CampaignArn
GetPersonalizedRankingLatency	接收 GetPersonalizedRanking API 调用与发送推荐（不包括 4xx 和 5xx 错误）之间的时间。 尺寸：CampaignArn 单位：毫秒

指标	说明
GetRecommendations	GetRecommendations API 调用是否成功。使用sum统计数据查看成功 GetRecommendations API 调用的总数。此指标不支持其他统计信息。 尺寸：CampaignArn
GetRecommendations 4xxErrors	4xx HTTP 响应代码返回的 GetRecommendations API 调用的数量。 尺寸：CampaignArn
GetRecommendations5xxErrors	5xx HTTP 响应代码返回的 GetRecommendations API 调用的数量。 维度： CampaignArn
GetRecommendationsLatency	接收 GetRecommendations API 调用与发送推荐（不包括 4xx 和 5xx 错误）之间的时间。 尺寸：CampaignArn 单位：毫秒
PutEventsRequests	成功的 PutEvents API 调用的数量。 维度： DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents4xxErrors	4xx HTTP 响应代码返回的 PutEvents API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn, EventTrackerArn

指标	说明
PutEvents5xxErrors	5xx HTTP 响应代码返回的 PutEvents API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn, EventTrackerArn
PutEventLatency	完成 PutEvents API 调用（不包括 4xx 和 5xx 错误）花费的时间。 尺寸： DatasetGroupArn, DatasetArn, EventTrackerArn 单位：毫秒
PutItemsRequests	成功的 PutItems API 调用的数量。 维度： DatasetGroupArn, DatasetArn
PutItems4xxErrors	4xx HTTP 响应代码返回的 PutItems API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn
PutItems5xxErrors	5xx HTTP 响应代码返回的 PutItems API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn
PutItemsLatency	完成 PutItems API 调用（不包括 4xx 和 5xx 错误）花费的时间。 尺寸： DatasetGroupArn, DatasetArn 单位：毫秒
PutUsersRequests	成功的 PutUsers API 调用的数量。 维度： DatasetGroupArn, DatasetArn

指标	说明
PutUsers4xxErrors	4xx HTTP 响应代码返回的 PutUsers API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn
PutUsers5xxErrors	5xx HTTP 响应代码返回的 PutUsers API 调用的数量。 尺寸： DatasetGroupArn, DatasetArn
PutUsersLatency	完成 PutUsers API 调用（不包括 4xx 和 5xx 错误）花费的时间。 尺寸： DatasetGroupArn, DatasetArn 单位：毫秒

使用记录亚马逊个性化 API 调用 AWS CloudTrail

Amazon Person AWS CloudTrail alize 与一项服务集成，可记录用户、角色或 AWS 服务在 Amazon Personalize 中采取的操作。 CloudTrail 捕获一部分 Amazon Personalize 的 API 调用作为事件，包括来自亚马逊个性化控制台的调用和对亚马逊个性化 API 的代码调用。如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 Amazon Personalize 的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。通过收集的信息 CloudTrail，您可以确定向 Amazon Personalize 发出的请求、发出请求的 IP 地址、谁提出请求、何时提出请求以及其他详细信息。

要了解更多信息 CloudTrail，包括如何配置和启用它，请参阅AWS CloudTrail 用户指南。

Amazon Personalize 在 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当 Amazon Personalize 中出现支持的事件活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅使用事件历史查看 CloudTrail 事件。

要持续记录您的 AWS 账户中的事件，包括 Amazon Personalize 的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon

S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅:

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

Amazon Personalize 支持将每个操作（API 操作）作为事件 CloudTrail 记录在日志文件中。有关更多信息，请参阅 操作。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根凭证还是用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

示例：Amazon Personalize 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

以下示例显示了包 CloudTrail 含 ListDatasetGroups API 操作操作的日志条目。请注意，由于 ListDatasetGroups API 操作是一个不会更改状态的操作，因此`responseElements`响应为空。有关 CloudTrail 记录正文的更多信息，请参阅CloudTrail 记录内容。

```
{
    "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"
}
```

Amazon Personalize 的合规性验证

Third-party 作为多项合规计划的一部分，审计师对 Amazon Personalize 的安全 AWS 性和合规性进行评估。其中包括 SOC、PCI、HIPAA 等。

有关特定合规计划范围内的 AWS 服务列表，请参阅AWS 按合规计划划分的范围内AWS 服务（按合分）。有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您使用 Amazon Personalize 的合规性责任取决于您数据的敏感度、您公司的合规性目标以及适用的法律法规。 AWS 提供以下资源来帮助满足合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在上面部署以安全性和合规性为重点的基准环境的步骤。 AWS
- HIPAA 安全与合规架构白皮书 — 了解如何使用它 AWS 来运行受监管的敏感工作负载。 U.S《健康保险流通与责任法案》（HIPAA）。
- AWS 合AWS 规资源 — 此工作簿和指南集合可能适用于您的行业和所在地区。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub CSPM— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

Amazon Personalize 中的故障恢复能力

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。 AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

Amazon Personalize 利用 AWS 全球基础设施实现数据弹性。当您在某个 AWS 地区创建 Amazon Personalize 资源时，Amazon Personalize 会跨多个可用区域管理该资源的弹性和数据冗余。有关可以在其中创建 Amazon Personalize 资源的AWS 区域列表，请参阅《亚马逊网络服务通用参考》中的区域和终端节点。 AWS 有关 AWS 区域和可用区的更多信息，请参阅AWS 全球基础设施。

Amazon Personalize 中的基础设施安全性

作为一项托管服务，Amazon Personalize 受到 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 Amazon Personalize。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（短暂的）或 ECDHE（椭圆曲线短暂的 Diffie-Hellman）。 Diffie-Hellman大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

Amazon Personalize 和接口 VPC 端点 (AWS PrivateLink)

如果您使用亚马逊虚拟私有云（亚马逊 VPC）托管 AWS 资源，则可以在您的 VPC 和 Amazon Personalize 之间建立私有连接。此连接允许 Amazon Personalize 与您 VPC 上的资源通信，而不用经过公共 Internet。

Amazon VPC 用于在您定义的虚拟私有云 (VPC) 或虚拟网络中启动 AWS 资源。 AWS 服务 借助 VPC，您可以控制您的网络设置，如 IP 地址范围、子网、路由表和网络网关。使用 VPC 终端节点，AWS 网络负责处理您的 VPC 和之间的路由 AWS 服务。

要将您的 VPC 连接到 Amazon Personalize，请为 Amazon Personalize 定义一个接口 VPC 端点。接口端点是具有私有 IP 地址的弹性网络接口，用作发送到受支持的 AWS 服务的流量的入口点。端点提供与 Amazon Personalize 之间的可靠、可扩展的连接。它不需要 Internet 网关、网络地址转换 (NAT) 实例或 VPN 连接。有关更多信息，请参阅 Amazon VPC 用户指南中的什么是 Amazon VPC。

接口 VPC 终端节点由启用 AWS PrivateLink。该 AWS 技术 AWS 服务 通过使用带有私有 IP 地址的 elastic network interface 来实现两者之间的私密通信。

> **Note**
>
> 支持所有 Amazon Personalize 联邦信息处理标准 (FIPS) 终端节点。 AWS PrivateLink

主题

- 为 Amazon Personalize 创建接口 VPC 端点
- 为 Amazon Personalize 创建 VPC 端点策略

为 Amazon Personalize 创建接口 VPC 端点

您可以使用亚马逊 VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 Amazon Personalize 服务创建 VPC 终端节点。有关更多信息，请参阅 Amazon VPC 用户指南中的使用接口 VPC 终端节点访问 AWS 服务。

要为 Amazon Personalize 创建 VPC 端点，请为该服务选择以下项之一：您可以选择创建 ipv4、ipv6 或双堆栈端点。

- com.amazonaws。 *region*. 个性化
- com.amazonaws。 *region*. 个性化活动
- com.amazonaws。 *region*.personalize 运行时

如果为端点启用私有 DNS，则可以使用其对于该区域的默认 DNS 名称，向 Amazon Personalize 发送 API 请求，例如 `personalize.us-east-1.api.aws`。

为 Amazon Personalize 创建 VPC 端点策略

您可以为 VPC 端点附加控制对 Amazon Personalize 的访问的端点策略。该策略指定以下信息：

- 可执行操作的主体。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问。

示例：允许所有 Amazon Personalize 操作和 passRole 操作的 VPC 端点策略

当附加到端点时，此策略会授予对所有 Amazon Personalize 操作和 passRole 操作的访问权限。

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:*",
                "iam:PassRole"
            ],
            "Resource": "*"
        }
    ]
}
```

示例：允许 Amazon Personalize ListDatasets 操作的 VPC 端点策略

当关联到终端节点时，此政策授予访问列出的 Amazon Personalize ListDatasets 操作的权限。

```
{
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Allow",
            "Action": [
                "personalize:ListDatasets"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Personalize 端点和限额

以下各节包含有关 Amazon Personalize 准则、配额和端点的信息。对于可调整限额，您可以使用服务限额控制台请求增加限额。有关更多信息，请参阅 请求提高配额。

主题

- Amazon Personalize 端点和区域
- 合规
- 服务配额
- 请求提高配额

Amazon Personalize 端点和区域

有关按区域划分的 Amazon Personalize 端点的列表，请参阅《Amazon Web Services 一般参考》中的 AWS 区域和端点。

合规

有关 Amazon Personalize 合规性计划的信息，请参阅 AWS 合规性、AWS 合规性计划以及合规性计划范围内的AWS 服务。

服务配额

您的 AWS 账户具有以下 Amazon Personalize 配额。

资源	限额
Item interactions	
创建解决方案版本或推荐程序所需的最少唯一项目交互次数。对于自定义解决方案，在训练之前，在按事件类型或事件值进行筛选后，必须有这么多记录。	1000

资源	限额
对于 User-Personalization-v 2 和 Personalized-Ranking-v 2 个配方，模型在训练期间考虑的最大物品交互次数。	30 亿
对于除 User-Personalization-v 2 或 Personalized-Ranking-v 2 以外的所有领域用例和自定义配方，模型在训练期间考虑的最大项目交互次数。	5 亿（可调）
项目交互数据集中不同事件类型的最大数量加上可选元数据列的总数。	10
项目交互数据集中元数据列的最大数量，不包括保留字段。	5
分类数据和展示值的最大字符数。	1000
在完全导入模式下，每个数据集导入任务的最大批量项目交互数据量。	100 GB（如果模型考虑增加项目交互量，则增加到 1TB）
在增量导入模式下，每个数据集导入任务的最大批量项目交互数据量。	1 GB
在完全或增量导入模式下，每个数据集导入任务的最小项目交互记录数。	1000
Users	
创建域名推荐器或自定义解决方案版本所需的项目交互数据中唯一用户的最小数量，每个用户至少有 2 个项目互动。	25
在创建域名推荐工具或自定义解决方案版本之前，必须进行至少 2 次或更多项目互动的用户总数的最小百分比。	1%
用户数据集的最大元数据字段数。	25

资源	限额
USER_ID 数据值的最大字符数。	256
分类数据值的最大字符数。	1000 个字符
在完全导入模式下，每个数据集导入任务的最大批量用户数据量。	100 GB
在增量导入模式下，每个数据集导入任务的最大批量用户数据量。	1 GB
Items	
对于 User-Personalization-v Personalized-Ranking-v 2 或 2，模型在训练期间考虑的最大项目数。这些物品来自物品数据集和物品交互数据集。	5 百万
对于除 2 和 User-Personalization-v 2 之外的所有域用例和 Personalized-Ranking-v自定义配方，模型在训练和生成推荐期间考虑的最大项目数。	750,000
Items 数据集的最大元数据字段数。	100
ITEM_ID 数据值的最大字符数。	256
分类和非类别字符串数据值的最大字符数。	1000 个字符
Items 数据集的最大文本字段数。	1
中文和日语文本数据值的最大字符数。	7000 个角色
所有其他语言的文本数据值的最大字符数。	2 万个字符
在批量导入模式下，每个数据集导入任务的最大批量项目数据量。	100 GB
在增量导入模式下，每个数据集导入任务的最大批量项目数据量。	1 GB

资源	限额
Actions	
模型在训练和生成推荐期间考虑的最大操作数。	1000
操作数据集的最大元数据字段数。	10
ACTION_ID 数据值的最大字符数。	256
分类数据值的最大字符数。	1000 个字符
在批量导入模式下，每个数据集导入任务的最大批量操作数据量。	100 GB
在增量导入模式下，每个数据集导入任务的最大批量操作数据量。	1 GB
Action interactions	
模型在训练期间考虑的最大操作交互数。	5 亿
操作交互数据集中的元数据列的最大数量，不包括保留字段。	5
在完全导入模式下，每个数据集导入任务的最大批量交互数据量。	100 GB（模型考虑的行动项目交互量增加后增加到 1TB）
在增量导入模式下，每个数据集导入任务的最大批量交互数据量。	1 GB
Individual record import APIs	
每个数据集组的最大`PutEvents` 请求速率。	1000 /秒
`PutEvents` 调用中的事件的最大数量。	10
事件的最大大小。	10 KB
每个数据集组的最大`PutActionInteractions` 请求速率。	1000 /秒

资源	限额
一次PutActionInteractions 通话中动作互动事件的最大数量。	10
动作互动事件的最大大小。	10 KB
每个数据集组的最大PutItems请求速率。	10/秒
一次PutItems通话中的最大项目数。	10
每个数据集组的最大PutUsers请求速率。	10/秒
一次PutUsers通话中的最大用户数。	10
每个数据集组的最大PutActions 请求速率。	10/秒
一次PutActions 通话中的最大用户数。	10
Legacy recipes	
HRNN-metadata 和 HRNN-ColdStart 配方的 “用户” 和 “项目” 数据集的最大合并数据量。	5 GB
HRNN-ColdStart 配方支持的最大冷启动项目数，用于训练模型（创建解决方案版本）。	80000
HRNN-ColdStart 配方训练模型（创建解决方案版本）所需的最小冷启动项目数量。	100
Filters	
每个数据集组的过滤器总数。	30（可调整）
过滤器中不同数据集字段的最大数量。	10
数据集组中所有筛选器中不同数据集字段的总数。	20
过滤器考虑的每种事件类型每位用户的最大项目交互次数。	100 次互动（可调整）

资源	限额
过滤器考虑的每种事件类型每位用户的最大操作交互次数。	300 个动作互动（可调）
GetRecommendations / GetPersonalizedRanking / GetActionRecommendations requests	
GetRecommendations 、GetActionRecommendations 和GetPersonalizedRanking 请求的最大交易速率。	2500 个/秒
每个活动每秒的最大 GetRecommendations 请求数。	500 个/秒
每个活动每秒的最大 GetActionRecommendations 请求数。	500 个/秒
每个活动每秒的最大 GetPersonalizedRanking 请求数。	500 /秒。
每个GetPersonalizedRanking 请求的最大元数据GetRecommendations 列数。	10
不含元数据的GetRecommendation 请求的最大推荐结果数。	500
包含元数据的GetRecommendation 请求的最大推荐结果数。	50
不含元数据的GetPersonalizedRanking 请求中可以排名的项目的最大数量。	500
包含元数据的GetPersonalizedRanking 请求中用于排名的项目的最大数量。	50
Metric attribution quotas	
指标归因的最大指标数量	10
唯一事件归因来源的最大数量	100

资源	限额
Batch inference jobs	
批量推理作业的最大输入文件数。	1000
批处理推理作业输入的最大大小。	1 GB
不带主题的批量推理作业每个输入文件的最大记录数。	5000 万
带有主题的批量推理作业每个输入文件的最大记录数。	100
Batch segment jobs	
批处理分段作业的最大输入文件数。	1000
批处理分段作业输入的最大大小。	1 GB
Item-Affinity 配方中每个输入文件的最大查询次数。	500
recipe 的每个输入文件的最大查询 Item-Attribute-Affinity数。	10
每个区段的最大用户数	5 百万
Data deletion jobs	
状态为 PENDING 的数据集组的最大数据删除任务数。	5（可调整）
一个或多个数据删除输入文件的最大总大小	100 MB

您的 AWS 账户在每个地区都有以下配额。

资源	限额
有效架构的总数。	500

资源	限额
有效数据集组的总数。	5（可调整）
待处理或进行中的数据集导入作业总数。	5
待处理或进行中的批量推理作业总数。	5（可调整）
待处理或进行中的批量细分作业总数。	5
待处理或进行中的解决方案版本总数。	20（可调整）

每个数据集组具有以下配额。

资源	限额
有效解决方案的总数。	10（可调整）
有效市场活动的总数。	5（可调整）
建议总数。	5
筛选器总数。	30（可调整）
所有筛选器中不同数据集字段的总数。	20
状态为 PENDING 的数据集组的数据删除作业总数。	5

请求提高配额

对于可调整限额，您可以使用服务限额控制台请求增加限额。以下 Amazon Personalize 限额可调整：

- 模型在训练期间考虑的最大物品交互数。
- 每个数据集组的有效活动数
- 活动数据集组数
- 每个数据集组的有效筛选条件数
- 每个数据集组的有效解决方案数

- 每次增量导入的数据量
- 一个筛选条件可以考虑的每用户、每事件类型的最大物品交互数
- 待处理或进行中的批量推理作业总数
- 待处理或进行中的解决方案版本总数
- PutEvents 或 PutActionInteraction 请求的最大速率

要请求增加限额，请使用服务限额控制台，并按照《服务限额用户指南》的请求增加限额部分的步骤进行操作。

API 参考

本节提供 Amazon Personalize API 操作的文档。有关按区域划分的 Amazon Personalize 端点的列表，请参阅《AWS 一般参考》中的 AWS 区域和端点。

主题

- 操作
- 数据类型
- 常见错误类型
- 常见参数

操作

Amazon Personalize 支持以下操作：

- 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

Amazon Personalize Events 支持以下操作：

- PutActionInteractions
- PutActions
- PutEvents
- PutItems

- PutUsers

Amazon Personalize Runtime 支持以下操作：

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

Amazon Personalize

Amazon Personalize 支持以下操作：

- 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

服务：Amazon Personalize

根据存储在 Amazon S3 中的物品或用户的列表生成批量建议，并将建议导出到 Amazon S3 桶。

要生成批量建议，请指定解决方案版本的 ARN 以及输入和输出数据的 Amazon S3 URI。对于用户个性化、热门物品和个性化排名解决方案，批量推理作业会为输入文件中的每个用户 ID 生成建议物品列表。对于相关物品解决方案，该作业会为输入文件中的每个物品 ID 生成建议物品列表。

有关更多信息，请参阅创建批量推理作业。

如果您使用 Similar-Items 配方，亚马逊个性化可以在批量推荐中添加描述性主题。要生成主题，请将作业的模式设置为 THEME_GENERATION，并指定在输入数据中包含物品名称的字段的名称。

有关生成主题的更多信息，请参阅内容生成器中带有主题的批量建议。

您无法使用 Trending-Now 或 Next-Best-Action 配方获得批量推荐。

请求语法

```
{
   "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"
      }
   }
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

batchInferenceJobConfig

批量推理作业的配置详情。

类型：BatchInferenceJobConfig 对象

必需：否

batchInferenceJobMode

批量推理作业的模式。要为一组相似的物品生成描述性主题，请将作业模式设置为 THEME_GENERATION。如果您不想生成主题，请使用默认 BATCH_INFERENCE。

当您获得带有主题的批量建议时，将产生额外的成本。有关更多信息，请参阅 Amazon Personalize 定价。

类型：字符串

有效值：BATCH_INFERENCE | THEME_GENERATION

必需：否

filterArn

要应用于批量推理作业的筛选器的 ARN。有关使用筛选器的更多信息，请参阅筛选批量建议。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

jobInput

指向输入文件以作为建议的基础的 Amazon S3 路径。输入材料必须采用 JSON 格式。

类型：BatchInferenceJobInput 对象

是否必需：是

jobName

要创建的批量推理作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

jobOutput

指向存储作业输出的 Amazon S3 存储桶的路径。

类型：BatchInferenceJobOutput 对象

是否必需：是

numResults

要检索的建议的数量。

类型：整数

必需：否

roleArn

分别有权读取和写入您的输入和输出 Amazon S3 存储桶的 Amazon Identity and Access Management 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

是否必需：是

solutionVersionArn

将用于生成批量推理建议的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

tags

要应用于批量推理作业的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

themeGenerationConfig

对于主题生成作业，请指定物品数据集中包含每个物品名称的列的名称。

类型：ThemeGenerationConfig 对象

必需：否

响应语法

```
{
```

```
   "batchInferenceJobArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchInferenceJobArn

批量推理作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateBatchSegmentJob

服务：Amazon Personalize

创建批量细分作业。该操作最多可以处理 5000 万条记录，并且输入文件必须为 JSON 格式。有关更多信息，请参阅获取批量建议和用户细分。

请求语法

```
{
   "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"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

filterArn

要应用于批量细分作业的筛选器的 ARN。有关使用筛选器的更多信息，请参阅筛选批量建议。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

jobInput

用于生成批量细分作业的输入数据的 Amazon S3 路径。

类型：BatchSegmentJobInput 对象

是否必需：是

jobName

要创建的批量细分作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

jobOutput

存储作业输出的存储桶的 Amazon S3 路径。

类型：BatchSegmentJobOutput 对象

是否必需：是

numResults

批量细分作业为每行输入数据生成的预测用户数。每个细分的最大用户数为 500 万。

类型：整数

必需：否

roleArn

分别有权读取和写入您的输入和输出 Amazon S3 存储桶的 Amazon Identity and Access Management 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

是否必需：是

solutionVersionArn

您希望批量细分作业用于生成批量细分的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

tags

要应用于批量细分作业的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "batchSegmentJobArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchSegmentJobArn

批量细分作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateCampaign

服务：Amazon Personalize

> **Important**
>
> 当市场活动处于活动状态时，会产生市场活动费用。为避免产生不必要的费用，请务必在市场活动结束后删除它。有关市场活动成本的信息，请参阅 Amazon Personalize 定价。

创建可部署解决方案版本的市场活动。当客户调用GetRecommendations和 GetPersonalizedRanking API 时，会在请求中指定一个活动。

最低配置 TPS 和 Auto-Scaling

> **Important**
>
> 高 minProvisionedTPS 会增加您的成本。我们建议从 1 开始设置 minProvisionedTPS（默认值）。使用亚马逊 CloudWatch 指标跟踪您的使用情况，并在必要minProvisionedTPS时增加使用量。

在创建 Amazon Personalize 市场活动时，您可以为该市场活动指定预置的每秒最低事务数 (minProvisionedTPS)。这是 Amazon Personalize 为该市场活动预置的基准事务吞吐量。它会设置市场活动处于活动状态时的最低账单费用。事务是单个 GetRecommendations 或 GetPersonalizedRanking 请求。默认 minProvisionedTPS 为 1。

如果您的 TPS 增加到超过 minProvisionedTPS，则 Amazon Personalize 会自动纵向扩展和缩减预置容量，但从不会低于 minProvisionedTPS。增加容量时会有短时间延迟，这可能会导致事务丢失。当您的流量减少时，容量将恢复到 minProvisionedTPS。

您需要为预置的最低 TPS 付费，或者，如果您的请求超过 minProvisionedTPS，则需要为实际 TPS 付费。实际 TPS 是您发出的建议请求的总数。我们建议从低点开始minProvisionedTPS，使用亚马逊 CloudWatch 指标跟踪您的使用量，然后minProvisionedTPS根据需要增加使用量。

有关市场活动成本的更多信息，请参阅 Amazon Personalize 定价。

状态

市场活动可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

要获取活动状态，请致电DescribeCampaign。

> **Note**
>
> 等到市场活动的 `status` 为 `ACTIVE` 时，再向市场活动询问建议。

相关 API

- ListCampaigns
- DescribeCampaign
- UpdateCampaign
- DeleteCampaign

请求语法

```
{
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "rankingInfluence": {
         "string" : number
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
   "name": "string",
   "solutionVersionArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
```

```
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

campaignConfig

市场活动的配置详情。

类型：CampaignConfig 对象

必需：否

minProvisionedTPS

指定 Amazon Personalize 将支持的请求的最低预调配每秒事务数（建议）。高 minProvisionedTPS 会增加您的账单费用。我们建议从 1 开始设置 minProvisionedTPS（默认值）。使用亚马逊 CloudWatch 指标跟踪您的使用情况，并在必要minProvisionedTPS时增加使用量。

类型：整数

有效范围：最小值为 1。

必需：否

name

新市场活动的名称。市场活动名称在您的账户内必须是唯一的。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

solutionVersionArn

用于与市场活动一起部署的已训练模型的 Amazon 资源名称（ARN）。要指定解决方案的最新解决方案版本，请以 SolutionArn/$LATEST 格式指定解决方案的 ARN。如果True在中设置syncWithLatestSolutionVersion为，则必须使用此格式CampaignConfig。

如果要部署的模型不是解决方案的最新解决方案版本，请指定解决方案版本的 ARN。

有关自动市场活动更新的更多信息，请参阅启用自动市场活动更新。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

tags

要应用于市场活动的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "campaignArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

campaignArn

市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK

- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateDataDeletionJob

服务：Amazon Personalize

创建一个批量作业，该作业批量删除对 Amazon Personalize 数据集组中特定用户的所有引用。您可以在 Amazon S3 存储桶中包含用户 ID 的 CSV 文件中指定要删除的用户。作业完成后，Amazon Personalize 将不再根据用户的数据进行训练，在生成用户细分时也不再考虑这些用户。有关创建数据删除作业的更多信息，请参阅删除用户。

- 输入文件必须是一个 CSV 文件，其中包含一个列出用户 ID 的 USER_ID 列。有关准备 CSV 文件的更多信息，请参阅准备数据删除文件并将其上传到 Amazon S3。
- 要允许 Amazon Personalize 访问您的包含用户 ID 的输入 CSV 文件，您必须指定一个有权从数据来源读取数据的 IAM 服务角色。此角色需要存储桶及其内容的 `GetObject` 和 `ListBucket` 权限。这些权限与导入数据时的权限相同。有关授予对 Amazon S3 存储桶的访问权限的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

创建作业后，最多可能需要一天时间才能从数据集和模型中删除对用户的所有引用。在作业完成之前，Amazon Personalize 会继续在训练时使用这些数据。而且，如果您使用 User Segmentation 配方，则用户可能会出现在用户细分中。

状态

数据删除作业可能具有以下某一状态：

- PENDING > IN_PROGRESS > COMPLETED 或 FAILED

要获取数据删除任务的状态，请调用 DescribeDataDeletionJob API 操作并指定任务的亚马逊资源名称 (ARN)。如果状态为 FAILED，则响应中包含一个描述作业失败原因的 `failureReason` 键。

相关 API

- ListDataDeletionJobs
- DescribeDataDeletionJob

请求语法

```
{
   "datasetGroupArn": "string",
   "dataSource": {
```

```
      "dataLocation": "string"
   },
   "jobName": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

数据集组（包含您希望从中删除记录的数据集）的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

dataSource

包含待删除用户的用户 ID 列表的 Amazon S3 存储桶。

类型：DataSource 对象

是否必需：是

jobName

数据删除作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

roleArn

有权从 Amazon S3 数据来源读取的 IAM 角色的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

是否必需：是

tags

要应用于数据删除作业的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "dataDeletionJobArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

dataDeletionJobArn

数据删除作业的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2

- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateDataset

服务：Amazon Personalize

创建一个空数据集并将其添加到指定的数据集组。用于CreateDatasetImportJob将您的训练数据导入数据集。

有 5 种类型的数据集：

- 物品交互
- 物品
- 用户
- 操作交互
- 操作

每个数据集类型都有一个与所需字段类型关联的架构。训练模型（也称为创建解决方案）只需要 `Item interactions` 数据集。

数据集可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

要获取数据集的状态，请调用DescribeDataset。

相关 API

- CreateDatasetGroup
- ListDatasets
- DescribeDataset
- DeleteDataset

请求语法

```
{
   "datasetGroupArn": "string",
   "datasetType": "string",
   "name": "string",
```

```
   "schemaArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

要添加到数据集的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

datasetType

数据集的类型。

以下一个值（区分大小写）：

- 交互
- 物品
- 用户
- 操作
- Action_Interactions

类型：字符串

长度约束：最大长度为 256。

模式：`^[A-Za-z_]+$`

是否必需：是

name

数据集的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

schemaArn

与数据集关联的架构的 ARN。架构定义数据集字段。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

是否必需：是

tags

要应用于数据集的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "datasetArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetArn

数据集的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateDatasetExportJob

服务：Amazon Personalize

创建将数据从数据集导出到 Amazon S3 存储桶的作业。要允许 Amazon Personalize 导出训练数据，您必须指定一个与服务相关的 IAM 角色，以向 Amazon Personalize 提供对 Amazon S3 存储桶的 `PutObject` 权限。有关信息，请参阅《Amazon Personalize 开发人员指南》中的导出数据集。

状态

数据集导出作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

要获取导出任务的状态 DescribeDatasetExportJob，请调用并指定数据集导出任务的亚马逊资源名称 (ARN)。当状态显示为 ACTIVE 时，数据集导出完成。如果状态显示为 CREATE FAILED，则响应中包含一个描述作业失败原因的 `failureReason` 键。

请求语法

```
{
   "datasetArn": "string",
   "ingestionMode": "string",
   "jobName": "string",
   "jobOutput": {
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

包含要导出的数据的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

ingestionMode

要导出的数据，取决于您导入数据的方式。您可以选择仅导出使用BULK数据集导入任务导入PUT的数据，也可以选择仅导出增量导入的数据（使用控制台 PutUsers 和 PutItems 操作） PutEvents，或同时导ALL出两种类型。默认值为 PUT。

类型：字符串

有效值：`BULK | PUT | ALL`

必需：否

jobName

数据集导出作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

jobOutput

指向存储作业输出的 Amazon S3 存储桶的路径。

类型：DatasetExportJobOutput 对象

是否必需：是

roleArn

有权向输出 Amazon S3 存储桶添加数据的 IAM 服务角色的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

是否必需：是

tags

要应用于数据集导出作业的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "datasetExportJobArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetExportJobArn

数据集导出作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK

- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateDatasetGroup

服务：Amazon Personalize

创建空数据集组。数据集组是存放 Amazon Personalize 资源的容器。一个数据集组最多可以包含三个数据集，其中每个对应一种数据集：

- 物品交互
- 物品
- 用户
- 操作
- 操作交互

数据集组可以是域数据集组，您可以在其中指定域并使用诸如推荐器之类的预配置资源，也可以是自定义数据集组，在其中使用通过市场活动部署的自定义资源，例如带有解决方案版本的解决方案。如果您从域数据集组开始，则仍然可以添加自定义资源，例如解决方案和解决方案版本，这些版本使用自定义使用案例食谱进行训练，并通过市场活动部署。

数据集组可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING

要获取数据集组的状态，请致电DescribeDatasetGroup。如果状态显示为 CREATE FAILED，则响应中包含一个描述创建失败原因的 `failureReason` 键。

> **Note**
>
> 必须等到数据集组的 `status` 为 `ACTIVE`，然后才能向该组添加数据集。

您可以指定 AWS Key Management Service (KMS) 密钥来加密组中的数据集。如果您指定 KMS 密钥，则还必须包括有权访问该密钥的 AWS Identity and Access Management (IAM) 角色。

请求中需要数据集组 ARN 的 API

- CreateDataset
- CreateEventTracker

- CreateSolution

相关 API

- ListDatasetGroups
- DescribeDatasetGroup
- DeleteDatasetGroup

请求语法

```
{
   "domain": "string",
   "kmsKeyArn": "string",
   "name": "string",
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

domain

数据集组的域。要创建域数据集组，请指定域。您指定的域决定了数据集的默认架构和推荐器可用的使用案例。如果您未指定域，则可以创建一个自定义数据集组，其中包含您通过市场活动部署的解决方案版本。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

kmsKeyArn

用于加密数据集的 (KMS) 密钥的亚马逊资源名称 AWS Key Management Service (ARN)。

类型：字符串

长度约束：最大长度为 2048。

模式：`arn:aws.*:kms:.*:[0-9]{12}:key/.*`

必需：否

name

新数据集组的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

roleArn

有权访问 AWS Identity and Access Management (KMS) 密钥的 (IAM) 角色的 AWS Key Management Service ARN。仅当同时指定 KMS 密钥时，提供 IAM 角色才有效。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：否

tags

要应用于数据集组的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
```

```
    "datasetGroupArn": "string",
    "domain": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetGroupArn

新数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

domain

新域数据集组的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateDatasetImportJob

服务：Amazon Personalize

创建将训练数据从数据源（Amazon S3 存储桶）导入到 Amazon Personalize 数据集的作业。要允许 Amazon Personalize 导入训练数据，您必须指定一个有权从数据源读取数据的 IAM 服务角色，因为 Amazon Personalize 会生成您的数据副本并在内部进行处理。有关授予对 Amazon S3 存储桶的访问权限的信息，请参阅向 Amazon Personalize 授予访问 Amazon S3 资源的权限。

如果您已经创建了推荐器或通过市场活动部署了自定义解决方案版本，则新批量记录如何影响建议取决于您使用的域使用案例或食谱。有关更多信息，请参阅新数据如何影响实时建议。

> **Important**
>
> 默认情况下，数据集导入作业会替换您批量导入的数据集中的任何现有数据。要在不替换现有数据的情况下添加新记录，请在 CreateDatasetImportJob 操作中为导入模式指定增量。

状态

数据集导入作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

要获取导入任务的状态，请致电 DescribeDatasetImportJob，提供数据集导入任务的亚马逊资源名称 (ARN)。当状态显示为 ACTIVE 时，数据集导入完成。如果状态显示为 CREATE FAILED，则响应中包含一个描述作业失败原因的 `failureReason` 键。

> **Note**
>
> 导入需要耗费时间。您必须等到状态显示为 ACTIVE，然后才能使用数据集训练模型。

相关 API

- ListDatasetImportJobs
- DescribeDatasetImportJob

请求语法

```
{
   "datasetArn": "string",
   "dataSource": {
      "dataLocation": "string"
   },
   "importMode": "string",
   "jobName": "string",
   "publishAttributionMetricsToS3": boolean,
   "roleArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

接收导入数据的数据集的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

dataSource

包含要导入的训练数据的 Amazon S3 桶。

类型：DataSource 对象

是否必需：是

importMode

指定如何将新记录添加到现有数据集。默认导入模式为 FULL。如果您之前没有将批量记录导入数据集，则只能指定 FULL。

- 要覆盖数据集中的所有现有批量数据，请指定 FULL。您单独导入的数据不会被替换。
- 要将新记录附加到数据集中的现有数据中，请指定 INCREMENTAL。Amazon Personalize 会将所有具有相同 ID 的记录替换为新记录。

类型：字符串

有效值：FULL | INCREMENTAL

必需：否

jobName

数据集导入作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

publishAttributionMetricsToS3

如果您创建了指标归因，请指定是否将此导入作业的指标发布到 Amazon S3

类型：布尔值

必需：否

roleArn

有权读取 Amazon S3 数据来源的 IAM 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+

必需：否

tags

要应用于数据集导入作业的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "datasetImportJobArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetImportJobArn

数据集导入作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateEventTracker

服务：Amazon Personalize

创建事件跟踪器，供您在使用 PutEvents API 向指定数据集组添加事件数据时使用。

> **Note**
>
> 只有一个事件跟踪程序可以与数据集组关联。如果您使用相同的数据集组作为现有事件跟踪程序来调用 `CreateEventTracker`，将收到错误。

当您创建事件跟踪器时，响应包含一个跟踪 ID，您在使用该PutEvents操作时将其作为参数传递。然后，Amazon Personalize 会将事件数据附加到您在事件跟踪器中指定的数据集组的物品交互数据集中。

事件跟踪器可以处于以下某种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

要获取事件跟踪器的状态，请致电DescribeEventTracker。

> **Note**
>
> 在使用跟踪 ID 之前，事件跟踪器必须处于 ACTIVE 状态。

相关 API

- ListEventTrackers
- DescribeEventTracker
- DeleteEventTracker

请求语法

```
{
   "datasetGroupArn": "string",
   "name": "string",
```

```
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

接收事件数据的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

name

事件跟踪器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

tags

要应用于事件跟踪器的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "eventTrackerArn": "string",
   "trackingId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

eventTrackerArn

事件跟踪器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

trackingId

事件跟踪器的 ID。在向 PutEvents API 发出的请求中包含此 ID。

类型：字符串

长度约束：最大长度为 256。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateFilter

服务：Amazon Personalize

创建建议筛选器。有关更多信息，请参阅筛选建议和用户细分。

请求语法

```
{
   "datasetGroupArn": "string",
   "filterExpression": "string",
   "name": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

筛选器将属于的数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

filterExpression

筛选器表达式定义建议中包含或排除哪些物品。筛选器表达式必须遵循特定的格式规则。有关筛选器表达式结构和语法的信息，请参阅筛选器表达式。

类型：字符串

长度限制：最小长度为 1。最大长度为 2500。

是否必需：是

name

要创建的筛选器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

tags

要应用于筛选器的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "filterArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

filterArn

新筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包

- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateMetricAttribution

服务：Amazon Personalize

创建指标归因。指标归因会针对您导入 Amazon Personalize 的数据创建报告。根据您导入数据的方式，您可以在 Amazon CloudWatch 或 Amazon S3 中查看报告。有关更多信息，请参阅衡量建议的影响。

请求语法

```
{
   "datasetGroupArn": "string",
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "name": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

指标归因的目标数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

metrics

指标归因的指标属性列表。每个指标属性都指定了要跟踪的事件类型和函数。可用函数为 SUM() 或 SAMPLECOUNT()。对于 SUM() 函数，提供数据集类型（交互或物品）及要求和的列作为参数。例如 SUM (Items.PRICE)。

类型：MetricAttribute 对象数组

数组成员：最多 10 项。

是否必需：是

metricsOutputConfig

指标归因的输出配置详细信息。

类型：MetricAttributionOutput 对象

是否必需：是

name

指标归因的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

必需：是

响应语法

```
{
   "metricAttributionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metricAttributionArn

新指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateRecommender

服务：Amazon Personalize

通过您指定的食谱（域数据集组使用案例）创建推荐器。您可以为域名数据集组创建推荐人，并在提出请求时指定推荐者的亚马逊资源名称 (ARN)。 GetRecommendations

最低每秒建议请求数

> **Important**
>
> 高 minRecommendationRequestsPerSecond 会增加您的账单费用。我们建议从 1 开始设置 minRecommendationRequestsPerSecond（默认值）。使用亚马逊 CloudWatch 指标跟踪您的使用情况，并在必要minRecommendationRequestsPerSecond时增加使用量。

创建推荐器时，您可以配置推荐器的每秒最低建议请求数。每秒最低建议请求数 (minRecommendationRequestsPerSecond) 指定了 Amazon Personalize 预调配的基准建议请求吞吐量。默认最小值RecommendationRequestsPerSecond 为1。建议请求是单个 GetRecommendations 操作。请求吞吐量以每秒请求数来衡量，Amazon Personalize 使用每秒请求数来计算每小时请求数和推荐器使用量的价格。

如果您的每秒请求数增加到超过 minRecommendationRequestsPerSecond，则 Amazon Personalize 会自动纵向扩展和缩减预调配容量，但永远不会低于 minRecommendationRequestsPerSecond。增加容量时会有短时间延迟，这可能会导致请求丢失。

您的账单是每小时最低请求数（基于最小值RecommendationRequestsPerSecond）或实际请求数中的较大值。使用的实际请求吞吐量按一小时 requests/second 内的平均值计算。我们建议从默认值开始minRecommendationRequestsPerSecond，使用亚马逊 CloudWatch 指标跟踪您的使用量，然后minRecommendationRequestsPerSecond根据需要增加使用量。

状态

推荐器可处于以下一种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

要获得推荐人身份，请致电DescribeRecommender。

> **Note**
>
> 等到推荐器的 `status` 为 `ACTIVE` 时，再向推荐器询问建议。

相关 API

- ListRecommenders
- DescribeRecommender
- UpdateRecommender
- DeleteRecommender

请求语法

```
{
   "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"
      }
   ]
```

```
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

推荐器目标域数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

name

推荐器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

是否必需：是

recipeArn

推荐器将使用的食谱的 Amazon 资源名称 (ARN)。对于推荐器，食谱是域数据集组使用案例。只能使用域数据集组使用案例来创建推荐器。有关使用案例的信息，请参阅选择推荐器使用案例。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

recommenderConfig

推荐器的配置详细信息。

类型：RecommenderConfig 对象

必需：否

tags

要应用于推荐器的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "recommenderArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recommenderArn

推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包

- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateSchema

服务：Amazon Personalize

从指定的架构字符串创建 Amazon Personalize 架构。您创建的架构必须采用 Avro JSON 格式。

Amazon Personalize 可识别三种架构变体。每种架构都与数据集类型相关联，并具有一组必填字段和关键字。如果要为域数据集组中的数据集创建架构，则提供域数据集组的域。您在调用时指定架构CreateDataset。

有关架构的更多信息，请参阅数据集和架构。

相关 API

- ListSchemas
- DescribeSchema
- DeleteSchema

请求语法

```
{
   "domain": "string",
   "name": "string",
   "schema": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

domain

架构的域。如果要为域数据集组中的数据集创建架构，请指定您在创建域数据集组时选择的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

name

架构的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

schema

采用 Avro JSON 格式的架构。

类型：字符串

长度限制：最大长度为 20000。

是否必需：是

响应语法

```
{
   "schemaArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

schemaArn

所创建架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateSolution

服务：Amazon Personalize

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

创建用于训练模型的配置（创建解决方案版本）。此配置包括用于模型训练的配方和可选的训练配置，例如训练中使用的列和特征转换参数。有关配置解决方案的更多信息，请参阅创建和配置解决方案。

默认情况下，新解决方案使用自动训练，每 7 天创建一个解决方案版本。您可以更改训练频率。解决方案处于活动状态后，自动解决方案版本创建会在一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。有关更多信息，请参阅配置自动训练。

要关闭自动训练，请将 `performAutoTraining` 设置为 false。如果您关闭自动训练，则必须通过调用该CreateSolutionVersion操作手动创建解决方案版本。

培训开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的亚马逊资源名称 (ARN)。要获取其状态，请使用 DescribeSolutionVersion.

训练完成后，您可以通过调用来评估模型的准确性GetSolutionMetrics。当您对解决方案版本感到满意时，可以使用进行部署CreateCampaign。该活动通过 GetRecommendations API向客户提供建议。

> **Note**
>
> Amazon Personalize 目前不支持配置用于解决方案超参数优化的 `hpoObjective`。

状态

解决方案可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

要获取解决方案的状态，请致电DescribeSolution。如果您使用手动训练，则在调用 `CreateSolutionVersion` 之前，状态必须为 ACTIVE。

相关 API

- UpdateSolution
- ListSolutions
- CreateSolutionVersion
- DescribeSolution
- DeleteSolution

- ListSolutionVersions
- DescribeSolutionVersion

请求语法

```
{
   "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"
         }
      ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

提供训练数据的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

eventType

当您有多个事件类型（使用 EVENT_TYPE 架构字段）时，此参数指定使用哪种事件类型（例如，“单击”或“点赞”）来训练模型。

如果没有提供 `eventType`，则 Amazon Personalize 将使用所有交互进行同等权重的训练，而不考虑类型。

类型：字符串

长度约束：最大长度为 256。

必需：否

name

解决方案的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

是否必需：是

performAutoML

> **Important**
>
> 我们建议不要启用自动机器学习。相反，请将您的使用案例与可用的 Amazon Personalize 配方匹配。有关更多信息，请参阅选择配方。

是否执行自动机器学习 (AutoML)。默认值为 `false`。对于这种情况，您必须指定 `recipeArn`。

设置为 `true` 时，Amazon Personalize 会分析您的训练数据并选择最佳 USER_PERSONALIZATION 食谱和超参数。在这种情况下，必须省略 `recipeArn`。Amazon Personalize 通过使用不同的超参数值来运行测试，以确定最佳食谱。与选择特定食谱相比，AutoML 可以延长训练过程。

类型：布尔值

必需：否

performAutoTraining

解决方案是否使用自动训练来创建新的解决方案版本（经过训练的模型）。默认设置为 `True`，解决方案每 7 天自动创建新的解决方案版本。在解决方案配置过程中，您可以在 `AutoTrainingConfig` 中指定 `schedulingExpression` 来更改训练频率。有关自动训练的更多信息，请参阅配置自动训练。

解决方案处于活动状态后，自动解决方案版本创建会在一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。

培训开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的亚马逊资源名称 (ARN)。要获取其状态，请使用 DescribeSolutionVersion.

类型：布尔值

必需：否

performHPO

是否对指定或所选食谱执行超参数优化 (HPO)。默认值为 `false`。

执行 AutoML 时，此参数始终为 `true`，您不应将其设置为 `false`。

类型：布尔值

必需：否

performIncrementalUpdate

是否对模型进行增量训练更新。启用后，这允许模型更频繁地从新数据中学习，而无需进行全面的重新训练，从而实现近乎实时的个性化设置。仅使用语义相似度配方的解决方案支持此参数。

类型：布尔值

必需：否

recipeArn

用于模型训练的配方的 Amazon 资源名称 (ARN)。当 `performAutoML` 为 false 时，这是必需的。有关不同的 Amazon Personalize 配方及其 ARN 的信息，请参阅选择配方。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

solutionConfig

解决方案的配置属性。将 `performAutoML` 设置为 true 时，Amazon Personalize 仅评估解决方案配置的 `autoMLConfig` 部分。

> **Note**
>
> Amazon Personalize 目前不支持配置 `hpoObjective`。

类型：SolutionConfig 对象

必需：否

tags

要应用于解决方案的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

响应语法

```
{
   "solutionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

solutionArn

解决方案的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包

- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

CreateSolutionVersion

服务：Amazon Personalize

训练或重新训练自定义数据集组中的活动解决方案。使用CreateSolution操作创建解决方案，调用前必须处于 ACTIVE 状态CreateSolutionVersion。每当您调用此操作时，就会创建一个新版本的解决方案。

状态

解决方案版本可以处于下列状态之一：

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING
- CREATE STOPPED

要获取版本的状态，请致电DescribeSolutionVersion。等到状态显示为 ACTIVE 后，再调用 CreateCampaign。

如果状态显示为 CREATE FAILED，则响应中包含一个描述作业失败原因的 failureReason 键。

相关 API

- ListSolutionVersions
- DescribeSolutionVersion
- ListSolutions
- CreateSolution
- DescribeSolution
- DeleteSolution

请求语法

```
{
   "name": "string",
   "solutionArn": "string",
```

```
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ],
   "trainingMode": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

name

解决方案版本的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

必需：否

solutionArn

包含训练配置信息的解决方案的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

是否必需：是

tags

要应用于解决方案版本的标签列表。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

必需：否

trainingMode

创建解决方案版本时要执行的训练的范围。默认值为 FULL。这将基于数据集组中数据集的全部训练数据，创建一个全新的模型。

如果使用 User-Personalization，则可以将训练模式指定为UPDATE。这会更新模型以考虑在建议中加入新物品。这并非全面的再训练。您仍然应该每周完成一次全面的再训练。如果您指定 UPDATE，则 Amazon Personalize 将停止解决方案版本的自动更新。要恢复更新，请创建一个将训练模式设置为 FULL 的新解决方案，然后将其部署到市场活动中。有关自动更新的更多信息，请参阅自动更新。

只有当您已经使用该UPDATE选项根据输入解决方案创建了有效的解决方案版本，并且输入解决方案已使用该User-Personalization配方或传统HRNN-Coldstart配方进行了训练时，才能使用该FULL选项。

类型：字符串

有效值：FULL | UPDATE | AUTOTRAIN

必需：否

响应语法

```
{
   "solutionVersionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

solutionVersionArn

新解决方案版本的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2

- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteCampaign

服务：Amazon Personalize

通过删除解决方案部署来移除市场活动。市场活动所依据的解决方案不会被删除，可以在需要时重新部署。无法再在GetRecommendations请求中指定已删除的广告活动。有关创建广告系列的信息，请参阅CreateCampaign。

请求语法

```
{
   "campaignArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

campaignArn

要删除的市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteDataset

服务：Amazon Personalize

删除数据集。如果关联的 DatasetImportJob 或 SolutionVersion 处于 CREATE PENDING 或 IN PROGRESS 状态，则您无法删除数据集。有关删除数据集的更多信息，请参阅删除数据集。

请求语法

```
{
   "datasetArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

要删除的数据集的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteDatasetGroup

服务：Amazon Personalize

删除数据集组。在删除数据集组前，您必须先删除以下内容：

- 所有关联的事件跟踪程序。
- 所有关联的解决方案。
- 数据集组中的所有数据集。

请求语法

```
{
   "datasetGroupArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

要删除的数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteEventTracker

服务：Amazon Personalize

删除事件跟踪器。不会将数据集从数据集组中删除。有关事件跟踪器的更多信息，请参阅CreateEventTracker。

请求语法

```
{
   "eventTrackerArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

eventTrackerArn

要删除的事件跟踪器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteFilter

服务：Amazon Personalize

删除筛选器。

请求语法

```
{
   "filterArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

filterArn

要删除的筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteMetricAttribution

服务：Amazon Personalize

创建指标归因。

请求语法

```
{
   "metricAttributionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

metricAttributionArn

指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteRecommender

服务：Amazon Personalize

停用并移除推荐器。无法再在GetRecommendations请求中指定已删除的推荐人。

请求语法

```
{
   "recommenderArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recommenderArn

要删除的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteSchema

服务：Amazon Personalize

删除 schema。在删除架构之前，必须删除引用该架构的所有数据集。有关架构的更多信息，请参阅 CreateSchema。

请求语法

```
{
   "schemaArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

schemaArn

要删除的架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DeleteSolution

服务：Amazon Personalize

删除解决方案的所有版本和 Solution 对象本身。在删除解决方案之前，您必须删除基于该解决方案的所有市场活动。要确定哪些活动正在使用该解决方案，请致电ListCampaigns并提供该解决方案的亚马逊资源名称 (ARN)。如果关联的 SolutionVersion 处于 CREATE PENDING 或 IN PROGRESS 状态，则您无法删除解决方案。有关解决方案的更多信息，请参阅CreateSolution。

请求语法

```
{
   "solutionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

solutionArn

要删除的解决方案的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeAlgorithm

服务：Amazon Personalize

描述给定的算法。

请求语法

```
{
   "algorithmArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

algorithmArn

要描述的算法的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

algorithm

算法属性的列表。

类型：Algorithm 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeBatchInferenceJob

服务：Amazon Personalize

获取批量推理作业的属性，包括名称、Amazon 资源名称 (ARN)、状态、输入和输出配置，以及用于生成建议的解决方案版本的 ARN。

请求语法

```
{
   "batchInferenceJobArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

batchInferenceJobArn

要描述的批量推理作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
         }
      }
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchInferenceJob

有关指定批量推理作业的信息。

类型：BatchInferenceJob 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeBatchSegmentJob

服务：Amazon Personalize

获取批量细分作业的属性，包括名称、Amazon 资源名称 (ARN)、状态、输入和输出配置，以及用于生成建议的解决方案版本的 ARN。

请求语法

```
{
   "batchSegmentJobArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

batchSegmentJobArn

要描述的批量细分作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchSegmentJob

有关指定批量细分作业的信息。

类型：BatchSegmentJob 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeCampaign

服务：Amazon Personalize

描述给定的市场活动，包括其状态。

市场活动可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

当 `status` 为 `CREATE FAILED` 时，响应将包含 `failureReason` 键，该键描述了原因。

有关广告系列的更多信息，请参阅CreateCampaign。

请求语法

```
{
   "campaignArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

campaignArn

市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

campaign

> **Note**
> 仅当市场活动至少接到一个 latestCampaignUpdate 电话时，才会返回 UpdateCampaign 字段。

市场活动的属性。

> **Note**
> 仅当市场活动至少接到一个 latestCampaignUpdate 电话时，才会返回 UpdateCampaign 字段。

类型：Campaign 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK

- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeDataDeletionJob

服务：Amazon Personalize

描述由创建的数据删除任务 CreateDataDeletionJob，包括任务状态。

请求语法

```
{
   "dataDeletionJobArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

dataDeletionJobArn

数据删除作业的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
   "dataDeletionJob": {
      "creationDateTime": number,
      "dataDeletionJobArn": "string",
      "datasetGroupArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "numDeleted": number,
      "roleArn": "string",
```

```
        "status": "string"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

dataDeletionJob

有关数据删除作业的信息，包括状态。

状态是以下值之一：

- PENDING
- 进行中
- COMPLETED
- FAILED

类型：DataDeletionJob 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2

- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeDataset

服务：Amazon Personalize

描述指定的数据集。有关数据集的更多信息，请参阅CreateDataset。

请求语法

```
{
   "datasetArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

要描述的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

dataset

数据集的属性列表。

类型：Dataset 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeDatasetExportJob

服务：Amazon Personalize

描述由创建的数据集导出任务 CreateDatasetExportJob，包括导出任务的状态。

请求语法

```
{
   "datasetExportJobArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetExportJobArn

要描述的数据集导出作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetExportJob

有关数据集导出作业的信息，包括状态。

状态是以下值之一：

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

类型：DatasetExportJob 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeDatasetGroup

服务：Amazon Personalize

描述给定的数据集组。有关数据集组的更多信息，请参阅CreateDatasetGroup。

请求语法

```
{
   "datasetGroupArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

要描述的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "datasetGroup": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "domain": "string",
      "failureReason": "string",
      "kmsKeyArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "roleArn": "string",
      "status": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetGroup

数据集组的属性列表。

类型：DatasetGroup 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK

- AWS Ruby V3 软件开发工具包

DescribeDatasetImportJob

服务：Amazon Personalize

描述由创建的数据集导入任务 CreateDatasetImportJob，包括导入任务状态。

请求语法

```
{
   "datasetImportJobArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetImportJobArn

要描述的数据集导入作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "datasetImportJob": {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetImportJobArn": "string",
      "dataSource": {
         "dataLocation": "string"
      },
      "failureReason": "string",
      "importMode": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "publishAttributionMetricsToS3": boolean,
```

```
        "roleArn": "string",
        "status": "string"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetImportJob

有关数据集导入作业的信息，包括状态。

状态是以下值之一：

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

类型：DatasetImportJob 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeEventTracker

服务：Amazon Personalize

描述事件跟踪器。响应包括事件跟踪器的 `trackingId` 和 `status`。有关事件跟踪器的更多信息，请参阅CreateEventTracker。

请求语法

```
{
   "eventTrackerArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

eventTrackerArn

要描述的事件跟踪器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "eventTracker": {
      "accountId": "string",
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "eventTrackerArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string",
      "trackingId": "string"
   }
```

```
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

eventTracker

描述事件跟踪器的对象。

类型：EventTracker 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包

- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeFeatureTransformation

服务：Amazon Personalize

描述给定的特征转换。

请求语法

```
{
   "featureTransformationArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

featureTransformationArn

要描述的特征转换的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "featureTransformation": {
      "creationDateTime": number,
      "defaultParameters": {
         "string" : "string"
      },
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

featureTransformation

房 FeatureTransformation 产清单。

类型：FeatureTransformation 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK

- AWS Ruby V3 软件开发工具包

DescribeFilter

服务：Amazon Personalize

描述筛选器的属性。

请求语法

```
{
   "filterArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

filterArn

要描述的筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
   "filter": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "filterArn": "string",
      "filterExpression": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

filter

筛选器的详细信息。

类型：Filter 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK

- AWS Ruby V3 软件开发工具包

DescribeMetricAttribution

服务：Amazon Personalize

描述指标归因。

请求语法

```
{
   "metricAttributionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

metricAttributionArn

指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "metricAttribution": {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "metricAttributionArn": "string",
      "metricsOutputConfig": {
         "roleArn": "string",
         "s3DataDestination": {
            "kmsKeyArn": "string",
            "path": "string"
         }
```

```
        },
        "name": "string",
        "status": "string"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metricAttribution

指标归因的详细信息。

类型：MetricAttribution 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包

- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeRecipe

服务：Amazon Personalize

描述食谱。

一个食谱包含三项：

- 训练模型的算法。
- 控制训练的超参数。
- 用于在训练前修改输入数据的特征转换信息。

Amazon Personalize 提供一组预定义的食谱。使用 CreateSolution API 创建解决方案时，您可以指定配方。CreateSolution使用指定配方和训练数据集中的算法训练模型。该解决方案作为活动部署后，可以使用 GetRecommendations API 提供建议。

请求语法

```
{
   "recipeArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recipeArn

要描述的食谱的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
```

```
   "recipe": {
      "algorithmArn": "string",
      "creationDateTime": number,
      "description": "string",
      "featureTransformationArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "recipeType": "string",
      "status": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recipe

描述食谱的对象。

类型：Recipe 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeRecommender

服务：Amazon Personalize

描述给定的推荐器，包括其状态。

推荐器可处于以下一种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

当 `status` 为 `CREATE FAILED` 时，响应将包含 `failureReason` 键，该键描述了原因。

创建或删除推荐器时，`modelMetrics` 键为空。

有关推荐人的更多信息，请参阅CreateRecommender。

请求语法

```
{
   "recommenderArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recommenderArn

要描述的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "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"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recommender

推荐器的属性。

类型：Recommender 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeSchema

服务：Amazon Personalize

描述架构。有关架构的更多信息，请参阅 CreateSchema。

请求语法

```
{
   "schemaArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

schemaArn

要检索的架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "schema": {
      "creationDateTime": number,
      "domain": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "schema": "string",
      "schemaArn": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

schema

请求的架构。

类型：DatasetSchema 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeSolution

服务：Amazon Personalize

描述解决方案。有关解决方案的更多信息，请参阅CreateSolution。

请求语法

```
{
   "solutionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

solutionArn

要描述的解决方案的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
   "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"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

solution

描述解决方案的对象。

类型：Solution 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

DescribeSolutionVersion

服务：Amazon Personalize

描述解决方案的特定版本。有关解决方案的更多信息，请参阅 CreateSolution

请求语法

```
{
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

solutionVersionArn

解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
   "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"
        }
    }
  }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

solutionVersion

解决方案版本。

类型：SolutionVersion 对象

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

GetSolutionMetrics

服务：Amazon Personalize

获取指定解决方案版本的指标。

请求语法

```
{
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

solutionVersionArn

要获取指标的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "metrics": {
      "string" : number
   },
   "solutionVersionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metrics

解决方案版本的指标。有关更多信息，请参阅使用指标评估解决方案版本。

类型：字符串到双映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

solutionVersionArn

与请求中指定的解决方案版本 ARN 相同。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2

- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListBatchInferenceJobs

服务：Amazon Personalize

获取已在解决方案版本之外执行的批量推理作业的列表。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

每页返回的批量推理作业结果的最大数量。默认值是 100。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

用于请求下一页结果的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

必需：否

solutionVersionArn

创建批量推理作业所依据的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

响应语法

```
{
   "batchInferenceJobs": [
      {
         "batchInferenceJobArn": "string",
         "batchInferenceJobMode": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchInferenceJobs

包含返回的每项作业信息的列表。

类型：BatchInferenceJobSummary 对象数组

数组成员：最多 100 项。

nextToken

用于检索下一页结果的令牌。当没有更多结果可返回时，值为 null。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListBatchSegmentJobs

服务：Amazon Personalize

获取已在您指定的解决方案版本之外执行的批量细分作业的列表。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

每页返回的批量细分作业结果的最大数量。默认值是 100。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

用于请求下一页结果的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

必需：否

solutionVersionArn

批量细分作业用于生成批量细分的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

响应语法

```
{
   "batchSegmentJobs": [
      {
         "batchSegmentJobArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

batchSegmentJobs

包含返回的每项作业信息的列表。

类型：BatchSegmentJobSummary 对象数组

数组成员：最多 100 项。

nextToken

用于检索下一页结果的令牌。当没有更多结果可返回时，值为 null。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListCampaigns

服务：Amazon Personalize

返回使用给定解决方案的市场活动列表。如果未指定解决方案，则会列出与账户关联的所有市场活动。响应中将提供每个市场活动的属性，包括 Amazon 资源名称 (ARN)。有关广告系列的更多信息，请参阅CreateCampaign。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

要返回的市场活动的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上一次调用返回的代币，ListCampaigns用于获取下一组活动（如果存在）。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

必需：否

solutionArn

要列出市场活动的解决方案的 Amazon 资源名称 (ARN)。如果未指定解决方案，则会列出与账户关联的所有市场活动。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

响应语法

```
{
   "campaigns": [
      {
         "campaignArn": "string",
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

campaigns

市场活动列表。

类型：CampaignSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组市场活动（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListDataDeletionJobs

服务：Amazon Personalize

返回数据集组的数据删除作业列表，按创建时间排序，最新的作业排在最前面。如果未指定数据集组，则会列出与账户关联的所有数据删除作业。响应中将提供每个作业的属性，包括 Amazon 资源名称（ARN）。有关数据删除作业的更多信息，请参阅删除用户。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

要列出数据删除作业的数据集组的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的数据删除作业的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListDataDeletionJobs` 用于获取下一组作业（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "dataDeletionJobs": [
      {
         "creationDateTime": number,
         "dataDeletionJobArn": "string",
         "datasetGroupArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

dataDeletionJobs

数据删除作业列表。

类型：DataDeletionJobSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组数据删除作业（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListDatasetExportJobs

服务：Amazon Personalize

返回使用给定数据集的数据集导出作业列表。如果未指定数据集，则会列出与账户关联的所有数据集导出作业。响应中将提供每个数据集导出作业的属性，包括 Amazon 资源名称 (ARN)。有关数据集导出任务的更多信息，请参阅CreateDatasetExportJob。有关数据集的更多信息，请参阅CreateDataset。

请求语法

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

要列出数据集导出作业的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的数据集导出作业的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListDatasetExportJobs` 用于获取下一组数据集导出作业（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "datasetExportJobs": [
      {
         "creationDateTime": number,
         "datasetExportJobArn": "string",
         "failureReason": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetExportJobs

数据集导出作业列表。

类型：DatasetExportJobSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组数据集导出作业（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListDatasetGroups

服务：Amazon Personalize

返回数据集组的列表。响应中将提供每个数据集组的属性，包括 Amazon 资源名称 (ARN)。有关数据集组的更多信息，请参阅CreateDatasetGroup。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

要返回的数据集组的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 ListDatasetGroups 用于获取下一组数据集组（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
```

```
   "datasetGroups": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "domain": "string",
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetGroups

数据集组列表。

类型：DatasetGroupSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组数据集组（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

错误

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListDatasetImportJobs

服务：Amazon Personalize

返回使用给定数据集的数据集导入作业的列表。如果未指定数据集，则会列出与账户关联的所有数据集导入作业。响应中将提供每个数据集导入作业的属性，包括 Amazon 资源名称 (ARN)。有关数据集导入任务的更多信息，请参阅CreateDatasetImportJob。有关数据集的更多信息，请参阅CreateDataset。

请求语法

```
{
   "datasetArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

要列出数据集导入作业的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的数据集导入作业的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 ListDatasetImportJobs 用于获取下一组数据集导入作业（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "datasetImportJobs": [
      {
         "creationDateTime": number,
         "datasetImportJobArn": "string",
         "failureReason": "string",
         "importMode": "string",
         "jobName": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetImportJobs

数据集导入作业的列表。

类型：DatasetImportJobSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组数据集导入作业（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListDatasets

服务：Amazon Personalize

返回给定数据集组中包含的数据集列表。响应中将提供每个数据集的属性，包括 Amazon 资源名称 (ARN)。有关数据集的更多信息，请参阅CreateDataset。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

包含要列出的数据集的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的数据集的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListDatasets` 用于获取下一组数据集导入作业（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "datasets": [
      {
         "creationDateTime": number,
         "datasetArn": "string",
         "datasetType": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasets

Dataset 数据元数组。每个对象都提供元数据信息。

类型：DatasetSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组数据集（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListEventTrackers

服务：Amazon Personalize

返回与账户关联的事件跟踪器的列表。响应中将提供每个事件跟踪器的属性，包括 Amazon 资源名称 (ARN) 和跟踪 ID。有关事件跟踪器的更多信息，请参阅CreateEventTracker。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

用于筛选响应的数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的事件跟踪器的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListEventTrackers` 用于获取下一组事件跟踪器（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "eventTrackers": [
      {
         "creationDateTime": number,
         "eventTrackerArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

eventTrackers

事件跟踪器列表。

类型：EventTrackerSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组事件跟踪器（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListFilters

服务：Amazon Personalize

列出属于给定数据集组的所有筛选器。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

包含筛选器的数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的筛选器的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListFilters` 用于获取下一组筛选器（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "Filters": [
      {
         "creationDateTime": number,
         "datasetGroupArn": "string",
         "failureReason": "string",
         "filterArn": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Filters

返回的筛选器的列表。

类型：FilterSummary 对象数组

数组成员：最多 100 项。

nextToken

用于获取下一组筛选器（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListMetricAttributionMetrics

服务：Amazon Personalize

列出指标归因的指标。

请求语法

```
{
   "maxResults": number,
   "metricAttributionArn": "string",
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

在一页结果中返回的指标的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

metricAttributionArn

要检索其属性的指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

nextToken

指定上次请求检索下一页结果时的分页令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "metrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metrics

指定指标归因的指标。

类型：MetricAttribute 对象数组

数组成员：最多 10 项。

nextToken

指定上次 ListMetricAttributionMetricsResponse 请求检索下一页结果时的分页令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListMetricAttributions

服务：Amazon Personalize

列出指标归因。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

指标归因的数据集组 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

在一页结果中返回的指标归因的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

指定上次请求检索下一页结果时的分页令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "metricAttributions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "metricAttributionArn": "string",
         "name": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metricAttributions

指标归因列表。

类型：MetricAttributionSummary 对象数组

数组成员：最多 100 项。

nextToken

指定上次请求检索下一页结果时的分页令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListRecipes

服务：Amazon Personalize

返回可用食谱列表。响应中将提供每个食谱的属性，包括食谱的 Amazon 资源名称 (ARN)。

请求语法

```
{
   "domain": "string",
   "maxResults": number,
   "nextToken": "string",
   "recipeProvider": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

domain

按域数据集组的域筛选返回的食谱。响应中仅包含该域的食谱（域数据集组使用案例）。如果不指定域，则返回所有食谱。

类型：字符串

有效值：ECOMMERCE | VIDEO_ON_DEMAND

必需：否

maxResults

要返回的食谱的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 ListRecipes 用于获取下一组食谱（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

recipeProvider

默认值为 SERVICE。

类型：字符串

有效值：SERVICE

必需：否

响应语法

```
{
   "nextToken": "string",
   "recipes": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "status": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

nextToken

用于获取下一组食谱的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

recipes

可用食谱的列表。

类型：RecipeSummary 对象数组

数组成员：最多 100 项。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListRecommenders

服务：Amazon Personalize

返回给定域数据集组中的推荐器列表。如果未指定域数据集组，则会列出与账户关联的所有推荐器。响应中将提供每个推荐器的属性，包括 Amazon 资源名称 (ARN)。有关推荐人的更多信息，请参阅CreateRecommender。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

要列出推荐器的域数据集组的 Amazon 资源名称 (ARN)。如果未指定域数据集组，则会列出与账户关联的所有推荐器。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的推荐器的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListRecommenders` 用于获取下一组推荐器（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "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"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

nextToken

用于获取下一组推荐器（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

recommenders

推荐器的列表。

类型：RecommenderSummary 对象数组

数组成员：最多 100 项。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListSchemas

服务：Amazon Personalize

返回与账户关联的架构的列表。响应中将提供每个架构的属性，包括 Amazon 资源名称 (ARN)。有关架构的更多信息，请参阅 CreateSchema。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

要返回的架构的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 ListSchemas 用于获取下一组架构（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
```

```
   "nextToken": "string",
   "schemas": [
      {
         "creationDateTime": number,
         "domain": "string",
         "lastUpdatedDateTime": number,
         "name": "string",
         "schemaArn": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

nextToken

用于获取下一组架构（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：`\p{ASCII}{0,1500}`

schemas

架构列表。

类型：DatasetSchemaSummary 对象数组

数组成员：最多 100 项。

错误

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListSolutions

服务：Amazon Personalize

返回给定数据集组中的解决方案列表。如果未指定数据集组，则会列出与账户关联的所有解决方案。响应中将提供每个解决方案的属性，包括 Amazon 资源名称 (ARN)。有关解决方案的更多信息，请参阅CreateSolution。

请求语法

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetGroupArn

数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

maxResults

要返回的解决方案的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 `ListSolutions` 用于获取下一组解决方案（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

响应语法

```
{
   "nextToken": "string",
   "solutions": [
      {
         "creationDateTime": number,
         "lastUpdatedDateTime": number,
         "name": "string",
         "recipeArn": "string",
         "solutionArn": "string",
         "status": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

nextToken

用于获取下一组解决方案（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

solutions

当前解决方案的列表。

类型：SolutionSummary 对象数组

数组成员：最多 100 项。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListSolutionVersions

服务：Amazon Personalize

返回给定解决方案的解决方案版本列表。如果未指定解决方案，则会列出与账户关联的所有解决方案版本。响应中将提供每个解决方案版本的属性，包括 Amazon 资源名称 (ARN)。

请求语法

```
{
   "maxResults": number,
   "nextToken": "string",
   "solutionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

maxResults

要返回的解决方案版本的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

nextToken

上次调用 ListSolutionVersions 用于获取下一组解决方案版本（如果存在）时返回的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

必需：否

solutionArn

解决方案的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

响应语法

```
{
   "nextToken": "string",
   "solutionVersions": [
      {
         "creationDateTime": number,
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "solutionVersionArn": "string",
         "status": "string",
         "trainingMode": "string",
         "trainingType": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

nextToken

用于获取下一组解决方案版本（如果存在）的令牌。

类型：字符串

长度限制：最大长度为 1500。

模式：\p{ASCII}{0,1500}

solutionVersions

描述版本属性的解决方案版本列表。

类型：SolutionVersionSummary 对象数组

数组成员：最多 100 项。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

InvalidNextTokenException

令牌无效。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

ListTagsForResource

服务：Amazon Personalize

获取附加到资源的标签列表。

请求语法

```
{
   "resourceArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

resourceArn

资源的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
{
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

tags

资源标签。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK

- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

StartRecommender

服务：Amazon Personalize

启动处于 INACTIVE 状态的推荐器。启动推荐器不会创建任何新模型，但会恢复对推荐器的计费和自动再训练。

请求语法

```
{
   "recommenderArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recommenderArn

要启动的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "recommenderArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recommenderArn

您启动的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

StopRecommender

服务：Amazon Personalize

停止处于 ACTIVE 状态的推荐器。停止推荐器会停止对推荐器的计费和自动再训练。

请求语法

```
{
   "recommenderArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recommenderArn

要停止的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
   "recommenderArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recommenderArn

您停止的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

StopSolutionVersionCreation

服务：Amazon Personalize

停止创建处于 CREATE_PENDING 或 CREATE IN_PROGRESS 状态的解决方案版本。

根据解决方案版本的当前状态，解决方案版本状态会发生如下变化：

- CREATE_PENDING > CREATE_STOPPED

  或者

- CREATE_IN_PROGRESS > CREATE_STOPPING > CREATE_STOPPED

在停止创建解决方案版本之前，您需要为所有已完成的训练付费。解决方案版本一旦停止，就无法继续创建。

请求语法

```
{
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

solutionVersionArn

要停止创建的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

TagResource

服务：Amazon Personalize

将标签列表添加到资源。

请求语法

```
{
   "resourceArn": "string",
   "tags": [
      {
         "tagKey": "string",
         "tagValue": "string"
      }
   ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

resourceArn

资源的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

tags

要应用于资源的标签。有关更多信息，请参阅标记 Amazon Personalize 资源。

类型：Tag 对象数组

数组成员：最少 0 个物品。最多 200 项。

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagsException

您已超出您可以应用到此资源的最大标签数量。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UntagResource

服务：Amazon Personalize

删除附加到资源的指定标签。有关更多信息，请参阅从 Amazon Personalize 资源删除标记。

请求语法

```
{
   "resourceArn": "string",
   "tagKeys": [ "string" ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

resourceArn

资源的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

tagKeys

要删除的标签的键。

类型：字符串数组

数组成员：最少 0 个物品。最多 200 项。

长度限制：最小长度为 1。最大长度为 128。

模式：`^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

是否必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

TooManyTagKeysException

该请求包含的标签键多于可与资源关联的标签键（每个资源 50 个标签键）。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UpdateCampaign

服务：Amazon Personalize

更新市场活动，以使用现有市场活动部署再训练的解决方案版本，更改市场活动的 minProvisionedTPS，或修改市场活动的配置。例如，您可以将现有市场活动的 enableMetadataWithRecommendations 设置为 true。

要将市场活动更新为使用最新解决方案版本自动开始，请指定以下内容：

- 在 SolutionVersionArn 参数中以 SolutionArn/$LATEST 格式指定解决方案的 Amazon 资源名称（ARN）。
- 在 campaignConfig 中，将 syncWithLatestSolutionVersion 设置为 true。

要更新市场活动，市场活动状态必须为 ACTIVE 或 CREATE FAILED。使用该DescribeCampaign操作检查活动状态。

> **Note**
>
> 更新进行期间，您仍可从市场活动中获得建议。在最新的市场活动更新状态变为 Active 之前，市场活动将使用以前的解决方案版本和市场活动配置来生成建议。

有关更新市场活动的更多信息（包括代码示例），请参阅更新市场活动。有关市场活动的更多信息，请参阅创建市场活动。

请求语法

```
{
   "campaignArn": "string",
   "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "rankingInfluence": {
         "string" : number
      },
      "syncWithLatestSolutionVersion": boolean
   },
   "minProvisionedTPS": number,
```

```
   "solutionVersionArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

campaignArn

市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

campaignConfig

市场活动的配置详情。

类型：CampaignConfig 对象

必需：否

minProvisionedTPS

指定 Amazon Personalize 将支持的请求的最低预调配每秒事务数（建议）。高 `minProvisionedTPS` 会增加您的账单费用。我们建议从 1 开始设置 `minProvisionedTPS`（默认值）。使用亚马逊 CloudWatch 指标跟踪您的使用情况，并在必要`minProvisionedTPS`时增加使用量。

类型：整数

有效范围：最小值为 1。

必需：否

solutionVersionArn

要部署的新模型的 Amazon 资源名称（ARN）。要指定解决方案的最新解决方案版本，请以 `SolutionArn/$LATEST` 格式指定解决方案的 ARN。如果`True`在中设置`syncWithLatestSolutionVersion`为，则必须使用此格式CampaignConfig。

如果要部署的模型不是解决方案的最新解决方案版本，请指定解决方案版本的 ARN。

有关自动市场活动更新的更多信息，请参阅启用自动市场活动更新。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

响应语法

```
{
   "campaignArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

campaignArn

与请求中提供的市场活动 ARN 相同。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UpdateDataset

服务：Amazon Personalize

更新数据集，将其架构替换为新的或现有的架构。有关更多信息，请参阅替换数据集的架构。

请求语法

```
{
   "datasetArn": "string",
   "schemaArn": "string"
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

datasetArn

您要更新的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

schemaArn

您要使用的新架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：是

响应语法

```
{
```

```
   "datasetArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

datasetArn

您更新的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2

- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UpdateMetricAttribution

服务：Amazon Personalize

创建指标归因。

请求语法

```
{
   "addMetrics": [
      {
         "eventType": "string",
         "expression": "string",
         "metricName": "string"
      }
   ],
   "metricAttributionArn": "string",
   "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
         "kmsKeyArn": "string",
         "path": "string"
      }
   },
   "removeMetrics": [ "string" ]
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

addMetrics

向指标归因添加新的指标属性。

类型：MetricAttribute 对象数组

数组成员：最多 10 项。

必需：否

metricAttributionArn

要更新的指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

metricsOutputConfig

指标归因的输出配置。

类型：MetricAttributionOutput 对象

必需：否

removeMetrics

从指标归因中移除指标属性。

类型：字符串数组

数组成员：最多 10 项。

长度约束：最大长度为 256。

必需：否

响应语法

```
{
   "metricAttributionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

metricAttributionArn

已更新的指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceAlreadyExistsException

指定资源已经存在。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK

- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UpdateRecommender

服务：Amazon Personalize

更新推荐器以修改推荐器配置。如果您更新推荐器以修改训练中使用的列，则 Amazon Personalize 会自动开始对支持推荐器的模型进行全面的重新训练。更新完成后，您仍然可以从推荐器中获得建议。在更新完成之前，推荐器使用之前的配置。要跟踪此更新的状态，请使用 DescribeRecommender 操作中返回的 `latestRecommenderUpdate`。

请求语法

```
{
   "recommenderArn": "string",
   "recommenderConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
         "string" : "string"
      },
      "minRecommendationRequestsPerSecond": number,
      "trainingDataConfig": {
         "excludedDatasetColumns": {
            "string" : [ "string" ]
         },
         "includedDatasetColumns": {
            "string" : [ "string" ]
         }
      }
   }
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

recommenderArn

要修改的推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

recommenderConfig

推荐器的配置详细信息。

类型：RecommenderConfig 对象

是否必需：是

响应语法

```
{
   "recommenderArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

recommenderArn

与请求中提供的推荐器 Amazon 资源名称 (ARN) 相同。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

UpdateSolution

服务：Amazon Personalize

更新 Amazon Personalize 解决方案以使用不同的自动训练配置。更新解决方案时，您可以更改解决方案是否使用自动训练，也可以更改训练频率。有关更新解决方案的更多信息，请参阅更新解决方案。

解决方案更新可处于以下几种状态之一：

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

要获取解决方案更新的状态，请调用 DescribeSolution API 操作并在中查找状态`latestSolutionUpdate`。

请求语法

```
{
   "performAutoTraining": boolean,
   "performIncrementalUpdate": boolean,
   "solutionArn": "string",
   "solutionUpdateConfig": {
      "autoTrainingConfig": {
         "schedulingExpression": "string"
      },
      "eventsConfig": {
         "eventParametersList": [
            {
               "eventType": "string",
               "eventValueThreshold": number,
               "weight": number
            }
         ]
      }
   }
}
```

请求参数

请求接受采用 JSON 格式的以下数据。

performAutoTraining

解决方案是否使用自动训练来创建新的解决方案版本（经过训练的模型）。在解决方案配置过程中，您可以在 `AutoTrainingConfig` 中指定 `schedulingExpression` 来更改训练频率。

如果开启自动训练，则第一次自动训练将在解决方案更新完成后的一小时内开始。如果您在一小时内手动创建解决方案版本，则解决方案会跳过第一次自动训练。有关自动训练的更多信息，请参阅配置自动训练。

培训开始后，您可以通过 ListSolutionVersions API 操作获取解决方案版本的亚马逊资源名称 (ARN)。要获取其状态，请使用 DescribeSolutionVersion.

类型：布尔值

必需：否

performIncrementalUpdate

是否对模型进行增量训练更新。启用后，这允许模型更频繁地从新数据中学习，而无需进行全面的重新训练，从而实现近乎实时的个性化设置。仅使用语义相似度配方的解决方案支持此参数。

类型：布尔值

必需：否

solutionArn

要更新的解决方案的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

solutionUpdateConfig

解决方案的新配置详细信息。

类型：SolutionUpdateConfig 对象

必需：否

响应语法

```
{
```

```
   "solutionArn": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

solutionArn

与请求中提供的解决方案 Amazon 资源名称（ARN）相同。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

LimitExceededException

超出每秒请求次数的限制。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：400

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

Amazon Personalize Events

Amazon Personalize Events 支持以下操作：

- PutActionInteractions
- PutActions
- PutEvents
- PutItems
- PutUsers

PutActionInteractions

服务：Amazon Personalize Events

记录操作交互事件数据。操作交互事件是用户和操作之间的交互。例如，用户执行一项操作，例如注册会员计划或下载您的应用。

有关记录操作交互的更多信息，请参阅记录操作交互事件。有关操作数据集中的操作的更多信息，请参阅操作数据集。

请求语法

```
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"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

actionInteractions

会话中的操作交互事件列表。

类型：ActionInteraction 对象数组

数组成员：最少 1 个物品。最多 10 个物品。

是否必需：是

trackingId

您的操作交互事件跟踪器的 ID。当您创建操作交互数据集时，Amazon Personalize 会为您创建一个操作交互事件跟踪器。有关更多信息，请参阅操作交互事件跟踪器 ID。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

是否必需：是

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：409

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

PutActions

服务：Amazon Personalize Events

将一个或多个操作添加到操作数据集。有关更多信息，请参阅单独导入操作。

请求语法

```
POST /actions HTTP/1.1
Content-type: application/json

{
   "actions": [
      {
         "actionId": "string",
         "properties": "string"
      }
   ],
   "datasetArn": "string"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

actions

操作数据的列表。

类型：Action 对象数组

数组成员：最少 1 个物品。最多 10 个物品。

是否必需：是

datasetArn

要向其中添加一个或多个操作的操作数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：是

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：409

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

PutEvents

服务：Amazon Personalize Events

记录物品交互事件数据。有关更多信息，请参阅记录物品交互事件。

> **Note**
>
> 如果您使用 AWS Lambda 函数调用 PutEvents 操作，则函数的执行角色必须具有使用`Resource`元素`*`中的通配符执行`personalize:PutEvents`操作的权限。

请求语法

```
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"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

eventList

会话中的事件数据列表。

类型：Event 对象数组

数组成员：最少 1 个物品。最多 10 个物品。

是否必需：是

sessionId

与用户访问关联的会话 ID。当用户首次访问您的网站或使用您的应用程序时，您的应用程序会生成 sessionId。Amazon Personalize 在用户登录之前使用 sessionId 将事件与用户关联起来。有关更多信息，请参阅记录物品交互事件。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

是否必需：是

trackingId

事件的跟踪 ID。该 ID 是通过调用 CreateEventTracker API 生成的。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

是否必需：是

userId

与事件关联的用户。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：否

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

PutItems

服务：Amazon Personalize Events

向物品数据集添加一个或多个物品。有关更多信息，请参阅单独导入物品。

请求语法

```
POST /items HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "items": [
      {
         "itemId": "string",
         "properties": "string"
      }
   ]
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

datasetArn

您要添加一个或多个物品的物品数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

items

物品数据列表。

类型：Item 对象数组

数组成员：最少 1 个物品。最多 10 个物品。

是否必需：是

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：409

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

PutUsers

服务：Amazon Personalize Events

向用户数据集中添加一个或多个用户。有关更多信息，请参阅单独导入用户。

请求语法

```
POST /users HTTP/1.1
Content-type: application/json

{
   "datasetArn": "string",
   "users": [
      {
         "properties": "string",
         "userId": "string"
      }
   ]
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

datasetArn

您要添加一个或多个用户的用户数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

users

用户数据列表。

类型：User 对象数组

数组成员：最少 1 个物品。最多 10 个物品。

是否必需：是

响应语法

```
HTTP/1.1 200
```

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceInUseException

指定的资源正在使用中。

HTTP 状态代码：409

ResourceNotFoundException

找不到指定的资源。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

Amazon Personalize Runtime

Amazon Personalize Runtime 支持以下操作：

- GetActionRecommendations
- GetPersonalizedRanking
- GetRecommendations

GetActionRecommendations

服务：Amazon Personalize Runtime

返回按预测分数以降序排列的建议操作列表。如果您的自定义市场活动部署了使用 PERSONALIZED_ACTIONS 配方训练的解决方案版本，请使用 GetActionRecommendations API。

有关 PERSONALIZED_ACTIONS 配方的更多信息，请参阅 PERSONALIZED_ACTIONS 配方。有关获取操作建议的更多信息，请参阅获取操作建议。

请求语法

```
POST /action-recommendations HTTP/1.1
Content-type: application/json

{
   "campaignArn": "string",
   "filterArn": "string",
   "filterValues": {
      "string" : "string"
   },
   "numResults": number,
   "userId": "string"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

campaignArn

用于获取操作建议的市场活动的 Amazon 资源名称 (ARN)。此市场活动必须部署使用 PERSONALIZED_ACTIONS 配方训练的解决方案版本。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

filterArn

要应用于返回建议的筛选器的 ARN。有关更多信息，请参阅筛选建议。

使用此参数时，确保筛选器资源为 ACTIVE。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

filterValues

筛选建议时要使用的值。对于筛选器表达式中的每个占位符参数，提供参数名称（大小写相符）作为键，提供筛选器值作为对应的值。用英文逗号分隔一个参数的多个值。

对于使用 INCLUDE 元素来包含操作的筛选条件表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 EXCLUDE 元素来排除操作的表达式的筛选条件，您可以省略 filter-values。在本例中，Amazon Personalize 未使用表达式的这一部分来筛选建议。

有关更多信息，请参阅筛选建议和用户细分。

类型：字符串到字符串映射

映射条目：最多 25 项。

键长度限制：最大长度为 50。

键模式：[A-Za-z0-9_]+

值长度限制：最大长度为 1000。

必需：否

numResults

要返回的结果数量。默认值为 5。最大值为 100。

类型：整数

有效范围：最小值为 0。

必需：否

userId

要为其提供操作建议的用户的用户 ID。

类型：字符串

长度约束：最大长度为 256。

必需：否

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
   "actionList": [
      {
         "actionId": "string",
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

actionList

按预测分数降序排序的操作建议列表。列表中最多可以有 100 个操作。有关操作分数的信息，请参阅操作建议评分的工作原理。

类型：PredictedAction 对象数组

recommendationId

建议的 ID。

类型：字符串

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

指定的资源不存在。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

GetPersonalizedRanking

服务：Amazon Personalize Runtime

Re-ranks 为给定用户推荐的物品列表。将列表中的第一个物品视为用户最可能感兴趣的物品。

> **Note**
>
> 必须已使用 PERSONALIZED_RANKING 类型的配方创建支持市场活动的解决方案。

请求语法

```
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"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

campaignArn

用于生成个性化排名的市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

是否必需：是

context

获取建议时要使用的上下文元数据。上下文元数据包括在获取用户建议时可能相关的任何交互信息，例如用户的当前位置或设备类型。

类型：字符串到字符串映射

映射条目：最多 150 项。

键长度限制：最大长度为 150。

键模式：`[A-Za-z\d_]+`

值长度限制：最大长度为 1000。

必需：否

filterArn

您创建的用于在给定用户建议中包含物品或将物品排除在此类建议之外的筛选器的 Amazon 资源名称 (ARN)。有关更多信息，请参阅筛选建议。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

filterValues

筛选建议时要使用的值。对于筛选器表达式中的每个占位符参数，提供参数名称（大小写相符）作为键，提供筛选器值作为对应的值。用英文逗号分隔一个参数的多个值。

对于使用 `INCLUDE` 元素来包含物品的筛选器表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 `EXCLUDE` 元素排除物品的表达式的筛选器，您可以省略 `filter-values`。在这种情况下，Amazon Personalize 不会使用表达式的这一部分来筛选建议。

有关更多信息，请参阅筛选建议。

类型：字符串到字符串映射

映射条目：最多 25 项。

键长度限制：最大长度为 50。

键模式：[A-Za-z0-9_]+

值长度限制：最大长度为 1000。

必需：否

inputList

要排名的物品（按 itemId 排名）列表。如果某个物品未包含在训练数据集中，则会将该物品附加到重新排名的列表末尾。如果您在建议中包含元数据，则最大值为 50。否则，最大值为 500。

类型：字符串数组

长度约束：最大长度为 256。

是否必需：是

metadataColumns

如果您在创建或更新市场活动时在建议中启用了元数据，请从您的物品数据集中指定要包含在个性化排名中的元数据列。映射键为 ITEMS，值是物品数据集中的列名称列表。您可以提供的最大列数为 10。

有关为市场活动启用元数据的信息，请参阅在市场活动建议中启用元数据。

类型：字符串到字符串数组映射

映射条目：最多 1 个物品。

密钥长度限制：最大长度为 256。

数组成员：最多 99 个物品。

长度限制：最大长度为 150。

必需：否

userId

您希望市场活动提供个性化排名的用户。

类型：字符串

长度约束：最大长度为 256。

是否必需：是

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
   "personalizedRanking": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
      }
   ],
   "recommendationId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

personalizedRanking

按用户最可能感兴趣的顺序排列的物品列表。最大值为 500。

类型：PredictedItem 对象数组

recommendationId

建议的 ID。

类型：字符串

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

指定的资源不存在。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本
- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

GetRecommendations

服务：Amazon Personalize Runtime

返回推荐物品的列表。对于市场活动，必须提供市场活动的 Amazon 资源名称 (ARN)，所需的用户和物品输入取决于用于创建支持市场活动的解决方案的食谱类型，如下所示：

- USER_PERSONALIZATION - `userId` 必填项，未使用 `itemId`
- RELATED_ITEMS - `itemId` 必填项，未使用 `userId`

> **Note**
>
> 由利用 PERSONALIZED_RANKING 类型的食谱创建的解决方案支持的市场活动使用 GetPersonalizedRanking API。

对于推荐器，推荐器的 ARN 是必填项，所需的物品和用户输入取决于支持推荐器的使用案例（基于域的食谱）。有关使用案例要求的信息，请参阅选择推荐器使用案例。

请求语法

```
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"
}
```

URI 请求参数

该请求不使用任何 URI 参数。

请求正文

请求接受采用 JSON 格式的以下数据。

campaignArn

用于获取建议的市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

context

获取建议时要使用的上下文元数据。上下文元数据包括在获取用户建议时可能相关的任何交互信息，例如用户的当前位置或设备类型。

类型：字符串到字符串映射

映射条目：最多 150 项。

键长度限制：最大长度为 150。

键模式：`[A-Za-z\d_]+`

值长度限制：最大长度为 1000。

必需：否

filterArn

要应用于返回建议的筛选器的 ARN。有关更多信息，请参阅筛选建议。

使用此参数时，确保筛选器资源为 ACTIVE。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

filterValues

筛选建议时要使用的值。对于筛选器表达式中的每个占位符参数，提供参数名称（大小写相符）作为键，提供筛选器值作为对应的值。用英文逗号分隔一个参数的多个值。

对于使用 INCLUDE 元素来包含物品的筛选器表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 EXCLUDE 元素排除物品的表达式的筛选器，您可以省略 filter-values。在这种情况下，Amazon Personalize 不会使用表达式的这一部分来筛选建议。

有关更多信息，请参阅筛选建议和用户细分。

类型：字符串到字符串映射

映射条目：最多 25 项。

键长度限制：最大长度为 50。

键模式：[A-Za-z0-9_]+

值长度限制：最大长度为 1000。

必需：否

itemId

要提供建议的物品 ID。

RELATED_ITEMS 食谱类型为必填项。

类型：字符串

长度约束：最大长度为 256。

必需：否

metadataColumns

如果您在创建或更新市场活动或推荐器时在建议中启用了元数据，请从您的物品数据集中指定要包含在物品建议中的元数据列。映射键为 ITEMS，值是物品数据集中的列名称列表。您可以提供的最大列数为 10。

有关为市场活动启用元数据的信息，请参阅在市场活动建议中启用元数据。有关为推荐器启用元数据的信息，请参阅在推荐器建议中启用元数据。

类型：字符串到字符串数组映射

映射条目：最多 1 个物品。

密钥长度限制：最大长度为 256。

数组成员：最多 99 个物品。

长度限制：最大长度为 150。

必需：否

numResults

要返回的结果数量。默认值为 25。如果您在建议中包含元数据，则最大值为 50。否则，最大值为 500。

类型：整数

有效范围：最小值为 0。

必需：否

promotions

要应用于建议请求的推广。推广定义了其他业务规则，这些规则适用于可配置的建议物品子集。

类型：Promotion 对象数组

数组成员：最多 1 个物品。

必需：否

recommenderArn

用于获取建议的推荐器的 Amazon 资源名称 (ARN)。如果您创建了包含域使用案例推荐器的域数据集组，请提供推荐器 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

userId

要提供建议的用户 ID。

USER_PERSONALIZATION 食谱类型为必填项。

类型：字符串

长度约束：最大长度为 256。

必需：否

响应语法

```
HTTP/1.1 200
Content-type: application/json

{
   "itemList": [
      {
         "itemId": "string",
         "metadata": {
            "string" : "string"
         },
         "promotionName": "string",
         "reason": [ "string" ],
         "score": number
      }
   ],
   "recommendationId": "string"
```

```
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

itemList

按预测分数降序排序的建议列表。列表中最多可以有 500 个物品。

类型：PredictedItem 对象数组

recommendationId

建议的 ID。

类型：字符串

错误

InvalidInputException

为字段或参数提供有效值。

HTTP 状态代码：400

ResourceNotFoundException

指定的资源不存在。

HTTP 状态代码：404

另请参阅

有关在某个特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行接口 V2
- AWS .NET V4 软件开发工具包
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK 版本

- AWS Java V2 软件开发工具包
- AWS JavaScript V3 版 SDK
- AWS 适用于 Kotlin 的 SDK
- AWS PHP V3 软件开发工具包
- AWS 适用于 Python (Boto3) 的 SDK
- AWS Ruby V3 软件开发工具包

数据类型

Amazon Personalize 支持以下数据类型：

- 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

Amazon Personalize Events 支持以下数据类型：

- Action
- ActionInteraction
- Event

- Item
- MetricAttribution
- User

Amazon Personalize Runtime 支持以下数据类型：

- PredictedAction
- PredictedItem
- Promotion

Amazon Personalize

Amazon Personalize 支持以下数据类型：

- 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

服务：Amazon Personalize

描述自定义算法。

目录

algorithmArn

算法的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

algorithmImage

算法映像的 Docker 容器的 URI。

类型：AlgorithmImage 对象

必需：否

creationDateTime

算法的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

defaultHyperParameterRanges

指定默认超参数、其范围及其是否可调。可以在超参数优化 (HPO) 期间确定可调超参数的值。

类型：DefaultHyperParameterRanges 对象

必需：否

defaultHyperParameters

指定默认超参数。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

defaultResourceConfig

指定默认最大训练作业数和并行训练作业数。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

lastUpdatedDateTime

上次更新算法的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

算法的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

roleArn

角色的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

trainingInputMode

训练输入模式。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

AlgorithmImage

服务：Amazon Personalize

描述算法映像。

目录

dockerURI

算法映像的 Docker 容器的 URI。

类型：字符串

长度约束：最大长度为 256。

必需：是

name

算法映像的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

AutoMLConfig

服务：Amazon Personalize

当解决方案执行 AutoML（CreateSolution 中 `performAutoML` 为 ture）时，Amazon Personalize 会从指定列表中确定哪个食谱对给定指标进行优化。然后，Amazon Personalize 将该食谱用于解决方案。

目录

metricName

要优化的指标。

类型：字符串

长度约束：最大长度为 256。

必需：否

recipeList

候选食谱列表。

类型：字符串数组

数组成员：最多 100 项。

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

AutoMLResult

服务：Amazon Personalize

当解决方案执行 AutoML（`performAutoML` 在 CreateSolution 中为 true）时，指定最能优化指定指标的食谱。

目录

bestRecipeArn

最佳食谱的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

AutoTrainingConfig

服务：Amazon Personalize

当 `performAutoTraining` 为 true 时要使用的自动训练配置。

目录

schedulingExpression

指定自动训练新解决方案版本的频率。以 rate(value unit) 格式指定 rate 表达式。对于值，指定一个介于 1 和 30 之间的数字。对于单位，指定 day 或 days。例如，要每 5 天自动创建一个新的解决方案版本，请指定 `rate(5 days)`。默认为每 7 天进行一次训练。

有关自动训练的更多信息，请参阅创建和配置解决方案。

类型：字符串

长度限制：长度下限为 1。最大长度为 16。

模式：`rate\(\d+ days?\)`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchInferenceJob

服务：Amazon Personalize

包含批量推理作业的相关信息。

目录

batchInferenceJobArn

批量推理作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

batchInferenceJobConfig

批量推理作业的配置详细信息的字符串到字符串映射。

类型：BatchInferenceJobConfig 对象

必需：否

batchInferenceJobMode

作业的模式。

类型：字符串

有效值：`BATCH_INFERENCE | THEME_GENERATION`

必需：否

creationDateTime

创建批量推理作业的时间。

类型：时间戳

必需：否

failureReason

如果批量推理作业失败，则为其原因。

类型：字符串

必需：否

filterArn

批量推理作业中使用的筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

jobInput

指向用于生成批量推理作业的输入数据的 Amazon S3 路径。

类型：BatchInferenceJobInput 对象

必需：否

jobName

批量推理作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

jobOutput

包含批量推理作业生成的输出数据的 Amazon S3 存储桶。

类型：BatchInferenceJobOutput 对象

必需：否

lastUpdatedDateTime

上次更新批量推理作业的时间。

类型：时间戳

必需：否

numResults

批量推理作业生成的建议数量。此数字包括为失败的输入记录生成的错误消息数。

类型：整数

必需：否

roleArn

请求批量推理作业的 Amazon Identity and Access Management (IAM) 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：否

solutionVersionArn

创建批量推理作业所依据的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

批量推理作业的状态。状态是以下值之一：

- PENDING
- IN PROGRESS

- ACTIVE
- CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

themeGenerationConfig

作业的主题生成设置。

类型：ThemeGenerationConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchInferenceJobConfig

服务：Amazon Personalize

批量推理作业的配置详情。

内容

itemExplorationConfig

指定浏览配置超参数的字符串到字符串映射，包括 `explorationWeight` 和 `explorationItemAgeCutOff`，您希望用它来配置 Amazon Personalize 在推荐物品时使用的物品浏览量。请参见 User-Personalization。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

rankingInfluence

排名地图影响受欢迎程度和新鲜度值。对于每个关键字，指定一个介于 0.0 和 1.0 之间的数值，该数值确定该排名因子对最终推荐的影响程度。值接近 1.0 会使因子得到更大的权重，而接近 0.0 的值会降低其影响。如果未指定，则两者都默认为 0.0。

类型：字符串到双映射

有效密钥：`POPULARITY | FRESHNESS`

有效范围：最小值为 0。最大值为 1。

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK

- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

BatchInferenceJobInput

服务：Amazon Personalize

批量推理作业的输入配置。

目录

s3DataSource

包含输入数据的 Amazon S3 位置的 URI。Amazon S3 存储桶必须与您要调用的 API 端点位于同一区域。

类型：S3DataConfig 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchInferenceJobOutput

服务：Amazon Personalize

批量推理作业的输出配置参数。

目录

s3DataDestination

存储批量推理作业输出的 Amazon S3 存储桶的相关信息。

类型：S3DataConfig 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchInferenceJobSummary

服务：Amazon Personalize

BatchInferenceJob 的截断版本。ListBatchInferenceJobs 操作返回批量推理作业摘要列表。

目录

batchInferenceJobArn

批量推理作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

batchInferenceJobMode

作业的模式。

类型：字符串

有效值：`BATCH_INFERENCE | THEME_GENERATION`

必需：否

creationDateTime

创建批量推理作业的时间。

类型：时间戳

必需：否

failureReason

如果批量推理作业失败，则为其原因。

类型：字符串

必需：否

jobName

批量推理作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新批量推理作业的时间。

类型：时间戳

必需：否

solutionVersionArn

批量推理作业使用的解决方案版本的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

批量推理作业的状态。状态是以下值之一：

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchSegmentJob

服务：Amazon Personalize

包含批量细分作业的相关信息。

目录

batchSegmentJobArn

批量细分作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

creationDateTime

创建批量细分作业的时间。

类型：时间戳

必需：否

failureReason

如果批量细分作业失败，则为其原因。

类型：字符串

必需：否

filterArn

批量细分作业中使用的筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

jobInput

指向用于生成批量细分作业的输入数据的 Amazon S3 路径。

类型：BatchSegmentJobInput 对象

必需：否

jobName

批量细分作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

jobOutput

包含批量细分作业生成的输出数据的 Amazon S3 存储桶。

类型：BatchSegmentJobOutput 对象

必需：否

lastUpdatedDateTime

上次更新批量细分作业的时间。

类型：时间戳

必需：否

numResults

批量细分作业为每行输入数据生成的预测用户数。每个细分的最大用户数为 500 万。

类型：整数

必需：否

roleArn

请求批量细分作业的 Amazon Identity and Access Management (IAM) 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：否

solutionVersionArn

批量细分作业用于生成批量细分的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

批量细分作业的状态。状态是以下值之一：

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchSegmentJobInput

服务：Amazon Personalize

批量细分作业的输入配置。

目录

s3DataSource

Amazon S3 输入或输出存储桶的配置详细信息。

类型：S3DataConfig 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchSegmentJobOutput

服务：Amazon Personalize

批量细分作业的输出配置参数。

目录

s3DataDestination

Amazon S3 输入或输出存储桶的配置详细信息。

类型：S3DataConfig 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

BatchSegmentJobSummary

服务：Amazon Personalize

BatchSegmentJob 数据类型的截断版本。ListBatchSegmentJobs 操作返回批量细分作业摘要列表。

目录

batchSegmentJobArn

批量细分作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

creationDateTime

创建批量细分作业的时间。

类型：时间戳

必需：否

failureReason

如果批量细分作业失败，则为其原因。

类型：字符串

必需：否

jobName

批量细分作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新批量细分作业的时间。

类型：时间戳

必需：否

solutionVersionArn

批量细分作业用于生成批量细分的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

批量细分作业的状态。状态是以下值之一：

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Campaign

服务：Amazon Personalize

描述解决方案版本部署的对象。有关市场活动的更多信息，请参阅 CreateCampaign。

目录

campaignArn

市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

campaignConfig

市场活动的配置详情。

类型：CampaignConfig 对象

必需：否

creationDateTime

市场活动创建的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

failureReason

市场活动失败背后的原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次市场活动的更新日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

latestCampaignUpdate

提供市场活动更新的属性摘要。如需完整列表，请调用 DescribeCampaign API。

> **Note**
> 仅当市场活动至少接到一个 `latestCampaignUpdate` 电话时，才会返回 `UpdateCampaign` 字段。

类型：CampaignUpdateSummary 对象

必需：否

minProvisionedTPS

指定请求的最低预调配每秒事务量（建议）。高 `minProvisionedTPS` 会增加您的账单费用。我们建议从 1 开始设置 `minProvisionedTPS`（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 `minProvisionedTPS`。

类型：整数

有效范围：最小值为 1。

必需：否

name

活动的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

solutionVersionArn

市场活动使用的解决方案版本的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

市场活动的状态。

市场活动可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

CampaignConfig

服务：Amazon Personalize

市场活动的配置详情。

内容

enableMetadataWithRecommendations

是否为市场活动启用了带有建议的元数据。如果启用，则可以在建议请求中指定物品数据集中的列。Amazon Personalize 会在建议响应中返回每个物品的这些数据。有关为市场活动启用元数据的信息，请参阅在市场活动建议中启用元数据。

如果您在建议中启用元数据，则会产生额外费用。有关更多信息，请参阅 Amazon Personalize 定价。

类型：布尔值

必需：否

itemExplorationConfig

指定浏览配置超参数，包括 `explorationWeight` 和 `explorationItemAgeCutOff`，您希望用它来配置 Amazon Personalize 在推荐物品时使用的物品浏览量。仅当您的解决方案使用 User-Personalization 食谱时，才提供 `itemExplorationConfig` 数据。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

rankingInfluence

排名地图影响受欢迎程度和新鲜度值。对于每个关键字，指定一个介于 0.0 和 1.0 之间的数值，该数值确定该排名因子对最终推荐的影响程度。值接近 1.0 会使因子得到更大的权重，而接近 0.0 的值会降低其影响。如果未指定，则两者都默认为 0.0。

类型：字符串到双映射

有效密钥：POPULARITY | FRESHNESS

有效范围：最小值为 0。最大值为 1。

必需：否

syncWithLatestSolutionVersion

市场活动是否自动更新为使用解决方案的最新解决方案版本（经过训练的模型）。如果指定为 True，则必须为 SolutionVersionArn 参数指定解决方案的 ARN。ARN 的格式必须为 SolutionArn/$LATEST。默认设置为 False，您必须手动更新市场活动才能部署最新的解决方案版本。

有关自动市场活动更新的更多信息，请参阅启用自动市场活动更新。

类型：布尔值

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

CampaignSummary

服务：Amazon Personalize

提供活动属性摘要。如需完整列表，请调用 DescribeCampaign API。

目录

campaignArn

市场活动的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

creationDateTime

活动创建的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

failureReason

市场活动失败背后的原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新市场活动的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

活动的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

市场活动的状态。

市场活动可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

CampaignUpdateSummary

服务：Amazon Personalize

提供市场活动更新的属性摘要。如需完整列表，请调用 DescribeCampaign API。

目录

campaignConfig

市场活动的配置详情。

类型：CampaignConfig 对象

必需：否

creationDateTime

创建市场活动更新的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

failureReason

市场活动更新失败背后的原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新市场活动的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

minProvisionedTPS

指定 Amazon Personalize 将支持的请求的最低预调配每秒事务数（建议）。

类型：整数

有效范围：最小值为 1。

必需：否

solutionVersionArn

部署的解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

市场活动更新的状态。

市场活动更新可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

CategoricalHyperParameterRange

服务：Amazon Personalize

提供分类超参数的名称和范围。

目录

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

values

超参数的类别列表。

类型：字符串数组

数组成员：最多 100 项。

长度限制：最大长度为 1000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ContinuousHyperParameterRange

服务：Amazon Personalize

提供连续超参数的名称和范围。

目录

maxValue

超参数的最大允许值。

类型：双精度

有效范围：最小值为 -1000000。

必需：否

minValue

超参数的最小允许值。

类型：双精度

有效范围：最小值为 -1000000。

必需：否

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

DataDeletionJob

服务：Amazon Personalize

描述一项作业，该作业批量删除对 Amazon Personalize 数据集组中特定用户的所有引用。有关创建数据删除作业的信息，请参阅删除用户。

目录

creationDateTime

数据删除作业的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

dataDeletionJobArn

数据删除作业的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetGroupArn

作业要从中删除记录的数据集组的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

dataSource

描述了包含要上传到数据集的数据来源，或者要从 Amazon Personalize 删除的记录列表。

类型：DataSource 对象

必需：否

failureReason

如果数据删除作业失败，请提供原因。

类型：字符串

必需：否

jobName

数据删除作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新数据删除作业的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

numDeleted

已完成作业删除的记录数。

类型：整数

必需：否

roleArn

有权从 Amazon S3 数据来源读取的 IAM 角色的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：否

status

数据删除作业的状态。

数据删除作业可能具有以下某一状态：

- PENDING > IN_PROGRESS > COMPLETED 或 FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DataDeletionJobSummary

服务：Amazon Personalize

提供数据删除作业属性的摘要。如需完整列表，请调用 DescribeDataDeletionJob API 操作。

目录

creationDateTime

数据删除作业的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

dataDeletionJobArn

数据删除作业的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetGroupArn

作业已从中删除记录的数据集组的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果数据删除作业失败，请提供原因。

类型：字符串

必需：否

jobName

数据删除作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新数据删除作业的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

status

数据删除作业的状态。

数据删除作业可能具有以下某一状态：

- PENDING > IN_PROGRESS > COMPLETED 或 FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Dataset

服务：Amazon Personalize

提供数据集的元数据。

内容

creationDateTime

数据集的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetArn

您想要元数据的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetGroupArn

数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetType

下列值之一：

- 交互
- 物品

- 用户
- 操作
- Action_Interactions

类型：字符串

长度约束：最大长度为 256。

模式：^[A-Za-z_]+$

必需：否

lastUpdatedDateTime

显示数据集更新时间的时间戳。

类型：时间戳

必需：否

latestDatasetUpdate

描述数据集的最新更新。

类型：DatasetUpdateSummary 对象

必需：否

name

数据集的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

必需：否

schemaArn

关联架构的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

数据集的状态。

数据集可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

trackingId

操作交互数据集的事件跟踪器的 ID。您可以在 `PutActionInteractions` API 操作中指定跟踪器的 ID。Amazon Personalize 使用它来将新数据定向到数据集组中的操作交互数据集。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

DatasetExportJob

服务：Amazon Personalize

描述将数据集导出到 Amazon S3 存储桶的作业。有关更多信息，请参阅 CreateDatasetExportJob。

数据集导出作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

目录

creationDateTime

数据集导出作业的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetArn

要导出的数据集的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetExportJobArn

数据集导出作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果数据集导出作业失败，请提供原因。

类型：字符串

必需：否

ingestionMode

要导出的数据，取决于您导入数据的方式。您可以选择导出使用数据集导入作业导入的 BULK 数据、增量导入的 PUT 数据（使用控制台、PutEvents、PutUsers 和 PutItems 操作），或者同时导出这两种类型的 ALL。默认值为 PUT。

类型：字符串

有效值：`BULK | PUT | ALL`

必需：否

jobName

导出作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

jobOutput

指向存储作业输出的 Amazon S3 存储桶的路径。例如：

```
s3://bucket-name/folder-name/
```

类型：DatasetExportJobOutput 对象

必需：否

lastUpdatedDateTime

上次更新数据集导出作业状态的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

roleArn

有权向输出 Amazon S3 存储桶添加数据的 IAM 服务角色的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

数据集导出作业的状态。

数据集导出作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetExportJobOutput

服务：Amazon Personalize

数据集导出作业的输出配置参数。

目录

s3DataDestination

Amazon S3 输入或输出存储桶的配置详细信息。

类型：S3DataConfig 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetExportJobSummary

服务：Amazon Personalize

提供数据集导出作业的属性摘要。如需完整列表，请调用 DescribeDatasetExportJob API。

目录

creationDateTime

创建数据集导出作业的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetExportJobArn

数据集导出作业的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果数据集导出作业失败，则为其原因。

类型：字符串

必需：否

jobName

数据集导出作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新数据集导出作业状态的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

status

数据集导出作业的状态。

数据集导出作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetGroup

服务：Amazon Personalize

数据集组是相关数据集（物品交互、用户、物品、操作、操作交互）的集合。您可通过调用 CreateDatasetGroup 创建数据集组。然后通过调用 CreateDataset 创建一个数据集并将其添加到数据集组。数据集组用于通过调用 CreateSolution 创建和训练解决方案。数据集组只能包含每种数据集类型中的一种。

您可以指定 AWS Key Management Service (KMS) 密钥来加密组中的数据集。

目录

creationDateTime

数据集组的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetGroupArn

数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

domain

域数据集组的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

failureReason

如果创建数据集组失败，请提供原因。

类型：字符串

必需：否

kmsKeyArn

用于加密数据集的 AWS Key Management Service (KMS) 密钥的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 2048。

模式：`arn:aws.*:kms:.*:[0-9]{12}:key/.*`

必需：否

lastUpdatedDateTime

数据集组的上次更新日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

数据集组的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

roleArn

有权访问 AWS Key Management Service (KMS) 密钥的 AWS Identity and Access Management (IAM) 角色的 ARN。仅当同时指定 KMS 密钥时，提供 IAM 角色才有效。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：否

status

数据集组的当前状态。

数据集组可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetGroupSummary

服务：Amazon Personalize

提供数据集组的属性摘要。如需完整列表，请调用 DescribeDatasetGroup API。

目录

creationDateTime

创建数据集组的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetGroupArn

数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

domain

域数据集组的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

failureReason

如果创建数据集组失败，则为其原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新数据集组的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

数据集组的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

数据集组的状态。

数据集组可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetImportJob

服务：Amazon Personalize

描述将训练数据从数据来源（Amazon S3 桶）导入到 Amazon Personalize 数据集的作业。有关更多信息，请参阅 CreateDatasetImportJob。

数据集导入作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

目录

creationDateTime

数据集导入作业的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetArn

接收导入数据的数据集的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetImportJobArn

数据集导入作业的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

dataSource

包含要导入的训练数据的 Amazon S3 桶。

类型：DataSource 对象

必需：否

failureReason

如果数据集导入作业失败，请提供原因。

类型：字符串

必需：否

importMode

数据集导入作业用于导入新记录的导入模式。

类型：字符串

有效值：`FULL | INCREMENTAL`

必需：否

jobName

导入作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新数据集的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

publishAttributionMetricsToS3

作业是否向 Amazon S3 发布指标以进行指标归因。

类型：布尔值

必需：否

roleArn

有权读取 Amazon S3 数据来源的 IAM 角色的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

数据集导入作业的状态。

数据集导入作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetImportJobSummary

服务：Amazon Personalize

提供数据集导入作业的属性摘要。如需完整列表，请调用 DescribeDatasetImportJob API。

目录

creationDateTime

创建数据集导入作业的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetImportJobArn

数据集导入作业的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果数据集导入作业失败，则为其原因。

类型：字符串

必需：否

importMode

数据集导入作业用于更新数据集中的数据的导入模式。有关更多信息，请参阅更新现有批量数据。

类型：字符串

有效值：`FULL | INCREMENTAL`

必需：否

jobName

数据集导入作业的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

lastUpdatedDateTime

上次更新数据集导入作业状态的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

status

数据集导入作业的状态。

数据集导入作业可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetSchema

服务：Amazon Personalize

描述数据集的架构。有关架构的更多信息，请参阅 CreateSchema。

目录

creationDateTime

架构的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

domain

为域数据集组中的数据集创建的架构的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

lastUpdatedDateTime

上次更新架构的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

架构的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

schema

架构。

类型：字符串

长度限制：最大长度为 20000。

必需：否

schemaArn

架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetSchemaSummary

服务：Amazon Personalize

提供数据集架构的属性摘要。如需完整列表，请调用 DescribeSchema API。

目录

creationDateTime

架构的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

domain

为域数据集组中的数据集创建的架构的域。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

lastUpdatedDateTime

上次更新架构的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

架构的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

schemaArn

架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DatasetSummary

服务：Amazon Personalize

提供数据集属性摘要。如需完整列表，请调用 DescribeDatasetAPI。

内容

creationDateTime

数据集的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetArn

数据集的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

datasetType

数据集类型。下列值之一：

- 交互
- 物品
- Users
- 事件-交互

类型：字符串

长度约束：最大长度为 256。

模式：`^[A-Za-z_]+$`

必需：否

lastUpdatedDateTime

上次更新数据集的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

数据集的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

数据集的状态。

数据集可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

DatasetUpdateSummary

服务：Amazon Personalize

描述数据集的更新。

目录

creationDateTime

数据集更新的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

failureReason

如果更新数据集失败，请提供原因。

类型：字符串

必需：否

lastUpdatedDateTime

数据集的上次更新日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

schemaArn

取代了数据集先前架构的架构的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

数据集更新的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DataSource

服务：Amazon Personalize

描述了包含要上传到数据集的数据来源，或者要从 Amazon Personalize 删除的记录列表。

目录

dataLocation

对于数据集导入作业，这是存储了要上传到数据集的数据的 Amazon S3 存储桶路径。对于数据删除作业，这是存储了待删除记录列表的 Amazon S3 存储桶路径。

例如：

```
s3://bucket-name/folder-name/fileName.csv
```

如果您的 CSV 文件位于 Amazon S3 存储桶的某个文件夹中，并且您希望导入作业或数据删除作业考虑多个文件，则可以指定该文件夹的路径。对于数据删除作业，Amazon Personalize 会使用该文件夹及任何子文件夹中的所有文件。使用以下语法，在文件夹名称后面加上 /：

```
s3://bucket-name/folder-name/
```

类型：字符串

长度约束：最大长度为 256。

模式：`(s3|http|https)://.+`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DefaultCategoricalHyperParameterRange

服务：Amazon Personalize

提供分类超参数的名称和默认范围，以及超参数是否可调。可以在超参数优化 (HPO) 期间确定可调超参数的值。

目录

isTunable

超参数是否可调。

类型：布尔值

必需：否

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

values

超参数的类别列表。

类型：字符串数组

数组成员：最多 100 项。

长度限制：最大长度为 1000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

DefaultContinuousHyperParameterRange

服务：Amazon Personalize

提供连续超参数的名称和默认范围，以及超参数是否可调。可以在超参数优化 (HPO) 期间确定可调超参数的值。

目录

isTunable

超参数是否可调。

类型：布尔值

必需：否

maxValue

超参数的最大允许值。

类型：双精度

有效范围：最小值为 -1000000。

必需：否

minValue

超参数的最小允许值。

类型：双精度

有效范围：最小值为 -1000000。

必需：否

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DefaultHyperParameterRanges

服务：Amazon Personalize

指定超参数及其默认范围。超参数可以是分类值、连续值或整数值。

目录

categoricalHyperParameterRanges

类别超参数及其默认范围。

类型：DefaultCategoricalHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

continuousHyperParameterRanges

连续超参数及其默认范围。

类型：DefaultContinuousHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

integerHyperParameterRanges

整数值超参数及其默认范围。

类型：DefaultIntegerHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

DefaultIntegerHyperParameterRange

服务：Amazon Personalize

提供整数值超参数的名称和默认范围，以及超参数是否可调。可以在超参数优化 (HPO) 期间确定可调超参数的值。

目录

isTunable

表示超参数是否可调。

类型：布尔值

必需：否

maxValue

超参数的最大允许值。

类型：整数

有效范围：最大值为 1000000。

必需：否

minValue

超参数的最小允许值。

类型：整数

有效范围：最小值为 -1000000。

必需：否

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EventParameters

服务：Amazon Personalize

描述用于创建解决方案的事件参数。

目录

eventType

创建解决方案时要考虑的事件类型的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

eventValueThreshold

事件类型的阈值。只有值大于或等于此阈值的事件才会被考虑用于创建解决方案。

类型：双精度

必需：否

weight

事件类型的权重。权重越高意味着事件类型对创建的解决方案越重要。

类型：双精度

有效范围：最小值为 0。最大值为 1。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EventsConfig

服务：Amazon Personalize

描述用于创建解决方案的事件配置。

目录

eventParametersList

事件参数列表，其中包含事件类型及其事件值阈值和权重。

类型：EventParameters 对象数组

数组成员：最多 10 项。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EventTracker

服务：Amazon Personalize

提供有关事件跟踪器的信息。

目录

accountId

拥有事件跟踪器的 AWS 账户。

类型：字符串

长度约束：最大长度为 256。

必需：否

creationDateTime

创建事件跟踪器的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

datasetGroupArn

接收事件数据的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

eventTrackerArn

事件跟踪器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

上次更新事件跟踪器的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

事件跟踪器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

事件跟踪器的状态。

事件跟踪器可以处于以下某种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

trackingId

事件跟踪器的 ID。在向 PutEvents API 发出的请求中包含此 ID。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EventTrackerSummary

服务：Amazon Personalize

提供事件跟踪器属性摘要。如需完整列表，请调用 DescribeEventTracker API。

目录

creationDateTime

创建事件跟踪器的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

eventTrackerArn

事件跟踪器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

上次更新事件跟踪器的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

事件跟踪器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

事件跟踪器的状态。

事件跟踪器可以处于以下某种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

FeatureTransformation

服务：Amazon Personalize

提供特征转换信息。特征转换是将原始输入数据修改为更适合模型训练的格式的过程。

目录

creationDateTime

特征转换的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

defaultParameters

提供特征转换的默认参数。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

featureTransformationArn

FeatureTransformation 对象的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

特征转换的上次更新日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

特征转换的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

特征转换的状态。

特征转换可以处于以下某种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

FieldsForThemeGeneration

服务：Amazon Personalize

主题生成的配置详细信息的字符串到字符串映射。

目录

itemName

用于在数据集中存储每个物品的名称的物品数据集列的名称。

类型：字符串

长度限制：最大长度为 150。

模式：[A-Za-z_][A-Za-z\d_]*

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Filter

服务：Amazon Personalize

包含有关建议筛选器的信息，包括其 ARN、状态和筛选器表达式。

目录

creationDateTime

创建筛选器的时间。

类型：时间戳

必需：否

datasetGroupArn

筛选器所属数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果筛选失败，请说明原因。

类型：字符串

必需：否

filterArn

筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

filterExpression

指定要从建议结果中筛选出来的物品交互类型。筛选器表达式必须遵循特定的格式规则。有关筛选器表达式结构和语法的信息，请参阅筛选器表达式。

类型：字符串

长度限制：长度下限为 1。最大长度为 2500。

必需：否

lastUpdatedDateTime

上次更新筛选器的时间。

类型：时间戳

必需：否

name

筛选器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

筛选器的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

FilterSummary

服务：Amazon Personalize

筛选器属性的简短摘要。

目录

creationDateTime

创建筛选器的时间。

类型：时间戳

必需：否

datasetGroupArn

筛选器所属数据集组的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果筛选失败，请说明原因。

类型：字符串

必需：否

filterArn

筛选器的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

上次更新筛选器的时间。

类型：时间戳

必需：否

name

筛选器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

筛选器的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

HPOConfig

服务：Amazon Personalize

描述超参数优化（HPO）的属性。

目录

algorithmHyperParameterRanges

超参数及其允许范围。

类型：HyperParameterRanges 对象

必需：否

hpoObjective

HPO 期间要优化的指标。

> **Note**
>
> Amazon Personalize 目前不支持配置 `hpoObjective`。

类型：HPOObjective 对象

必需：否

hpoResourceConfig

描述 HPO 的资源配置。

类型：HPOResourceConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

HPOObjective

服务：Amazon Personalize

超参数优化（HPO）期间要优化的指标。

> **Note**
>
> Amazon Personalize 目前不支持配置 `hpoObjective`。

目录

metricName

指标的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

metricRegex

用于在训练任务日志中查找指标的正则表达式。

类型：字符串

长度约束：最大长度为 256。

必需：否

type

指标的类型。有效值为 `Maximize` 和 `Minimize`。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

HPOResourceConfig

服务：Amazon Personalize

描述超参数优化（HPO）的资源配置。

目录

maxNumberOfTrainingJobs

创建解决方案版本时的最大训练任务数。maxNumberOfTrainingJobs 的最大值为 40。

类型：字符串

长度约束：最大长度为 256。

必需：否

maxParallelTrainingJobs

创建解决方案版本时的最大并行训练任务数。maxParallelTrainingJobs 的最大值为 10。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

HyperParameterRanges

服务：Amazon Personalize

指定超参数及其范围。超参数可以是分类值、连续值或整数值。

目录

categoricalHyperParameterRanges

分类超参数及其范围。

类型：CategoricalHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

continuousHyperParameterRanges

连续超参数及其范围。

类型：ContinuousHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

integerHyperParameterRanges

整数值超参数及其范围。

类型：IntegerHyperParameterRange 对象数组

数组成员：最多 100 项。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

IntegerHyperParameterRange

服务：Amazon Personalize

提供整数值超参数的名称和范围。

目录

maxValue

超参数的最大允许值。

类型：整数

有效范围：最大值为 1000000。

必需：否

minValue

超参数的最小允许值。

类型：整数

有效范围：最小值为 -1000000。

必需：否

name

超参数的名称。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

MetricAttribute

服务：Amazon Personalize

包含有关指标归因报告的指标的信息。有关更多信息，请参阅衡量建议的影响。

目录

eventType

指标事件类型。

类型：字符串

长度约束：最大长度为 256。

必需：是

expression

属性的表达式。可用函数为 SUM() 或 SAMPLECOUNT()。对于 SUM() 函数，提供数据集类型（交互或物品）及要求和的列作为参数。例如 SUM(Items.PRICE)。

类型：字符串

长度约束：最大长度为 256。

必需：是

metricName

指标名称。该名称可帮助您识别 Amazon CloudWatch 或 Amazon S3 中的指标。

类型：字符串

长度约束：最大长度为 256。

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

MetricAttribution

服务：Amazon Personalize

包含有关指标归因的信息。指标归因会针对您导入 Amazon Personalize 的数据创建报告。根据您导入数据的方式，您可以在 Amazon CloudWatch 或 Amazon S3 中查看报告。有关更多信息，请参阅衡量建议的影响。

目录

creationDateTime

指标归因的创建日期和时间。

类型：时间戳

必需：否

datasetGroupArn

指标归因的数据集组 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

指标归因的失败原因。

类型：字符串

必需：否

lastUpdatedDateTime

指标归因的上次更新日期和时间。

类型：时间戳

必需：否

metricAttributionArn

指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

metricsOutputConfig

指标归因的输出配置。

类型：MetricAttributionOutput 对象

必需：否

name

指标归因的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

指标归因的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

MetricAttributionOutput

服务：Amazon Personalize

指标归因的输出配置详细信息。

目录

roleArn

有权向输出 Amazon S3 存储桶添加数据并向 Amazon CloudWatch 添加指标的 IAM 服务角色的 Amazon 资源名称 (ARN)。有关更多信息，请参阅衡量建议的影响。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

必需：是

s3DataDestination

Amazon S3 输入或输出存储桶的配置详细信息。

类型：S3DataConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

MetricAttributionSummary

服务：Amazon Personalize

提供指标归因属性的摘要。如需完整列表，请调用 DescribeMetricAttribution。

目录

creationDateTime

指标归因的创建日期和时间。

类型：时间戳

必需：否

failureReason

指标归因的失败原因。

类型：字符串

必需：否

lastUpdatedDateTime

指标归因的上次更新日期和时间。

类型：时间戳

必需：否

metricAttributionArn

指标归因的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

name

指标归因的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

status

指标归因的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

OptimizationObjective

服务：Amazon Personalize

描述解决方案的其他目标，例如最大限度地提高流时长或增加收入。有关更多信息，请参阅优化解决方案。

目录

itemAttribute

物品数据集中与优化目标相关的数值元数据列。例如，VIDEO_LENGTH（可最大限度地提高流时长）或 PRICE（可最大限度地提高收入）。

类型：字符串

长度限制：长度下限为 1。最大长度为 150。

必需：否

objectiveSensitivity

指定 Amazon Personalize 如何平衡优化目标的重要性与相关性。

类型：字符串

有效值：`LOW | MEDIUM | HIGH | OFF`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Recipe

服务：Amazon Personalize

提供有关食谱的信息。每个食谱都提供了一种算法，供 Amazon Personalize 在您使用 CreateSolution 操作时在模型训练期间使用。

目录

algorithmArn

Amazon Personalize 用来训练模型的算法的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

creationDateTime

创建食谱的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

description

对食谱的描述。

类型：字符串

必需：否

featureTransformationArn

FeatureTransformation 对象的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

上次更新食谱的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

name

食谱的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

recipeArn

食谱的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

recipeType

下列值之一：

- PERSONALIZED_RANKING
- RELATED_ITEMS
- USER_PERSONALIZATION

类型：字符串

长度约束：最大长度为 256。

必需：否

status

食谱的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

RecipeSummary

服务：Amazon Personalize

提供食谱属性摘要。如需完整列表，请调用 DescribeRecipe API。

目录

creationDateTime

创建食谱的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

domain

食谱的域（如果食谱是域数据集组使用案例）。

类型：字符串

有效值：`ECOMMERCE | VIDEO_ON_DEMAND`

必需：否

lastUpdatedDateTime

上次更新食谱的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

食谱的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

recipeArn

食谱的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

食谱的状态。

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Recommender

服务：Amazon Personalize

描述域数据集组的建议生成器。您可以在域数据集组中为特定域使用案例（域食谱）创建推荐器，然后在 GetRecommendations 请求中指定推荐器。

目录

creationDateTime

推荐器的创建日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

datasetGroupArn

包含推荐器的域数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

failureReason

如果推荐器失败，请说明原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新推荐器的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

latestRecommenderUpdate

提供推荐器最新更新的摘要。

类型：RecommenderUpdateSummary 对象

必需：否

modelMetrics

提供评估指标，以帮助您确定推荐器的性能。有关更多信息，请参阅评估推荐器。

类型：字符串到双映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

必需：否

name

推荐器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

recipeArn

创建推荐器的食谱（域数据集组使用案例）的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

recommenderArn

推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

recommenderConfig

推荐器的配置详细信息。

类型：RecommenderConfig 对象

必需：否

status

推荐器的状态。

推荐器可处于以下一种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

RecommenderConfig

服务：Amazon Personalize

推荐器的配置详细信息。

目录

enableMetadataWithRecommendations

是否为推荐器启用了带有建议的元数据。如果启用，则可以在建议请求中指定物品数据集中的列。Amazon Personalize 会在建议响应中返回每个物品的这些数据。有关为推荐器启用元数据的信息，请参阅在推荐器建议中启用元数据。

如果您在建议中启用元数据，则会产生额外费用。有关更多信息，请参阅 Amazon Personalize 定价。

类型：布尔值

必需：否

itemExplorationConfig

指定浏览配置超参数，包括 `explorationWeight` 和 `explorationItemAgeCutOff`，您希望用它来配置 Amazon Personalize 在推荐物品时使用的物品浏览量。只有当您的推荐器为用户生成个性化建议（而非热门物品或类似物品）时，才提供 `itemExplorationConfig` 数据。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

minRecommendationRequestsPerSecond

指定 Amazon Personalize 将支持的请求的最低预调配建议每秒请求数。高 `minRecommendationRequestsPerSecond` 会增加您的账单费用。我们建议从 1 开始设置 `minRecommendationRequestsPerSecond`（默认值）。使用 Amazon CloudWatch 指标跟踪您的使用情况，并在必要时增加 `minRecommendationRequestsPerSecond`。

类型：整数

有效范围：最小值为 1。

必需：否

trainingDataConfig

指定创建域推荐器时要使用的训练数据配置。

类型：TrainingDataConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

RecommenderSummary

服务：Amazon Personalize

提供推荐器属性摘要。

目录

creationDateTime

推荐器的创建日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

datasetGroupArn

包含推荐器的域数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

lastUpdatedDateTime

上次更新推荐器的日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

name

推荐器的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

recipeArn

创建推荐器的食谱（域数据集组使用案例）的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

recommenderArn

推荐器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

recommenderConfig

推荐器的配置详细信息。

类型：RecommenderConfig 对象

必需：否

status

推荐器的状态。推荐器可处于以下一种状态：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

RecommenderUpdateSummary

服务：Amazon Personalize

提供推荐器更新的属性摘要。如需完整列表，请调用 DescribeRecommender API。

目录

creationDateTime

推荐器更新的创建日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

failureReason

如果推荐器更新失败，请说明原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新推荐器更新的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

recommenderConfig

推荐器更新的配置详细信息。

类型：RecommenderConfig 对象

必需：否

status

推荐器更新的状态。推荐器更新可处于以下一种状态：

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

S3DataConfig

服务：Amazon Personalize

Amazon S3 输入或输出存储桶的配置详细信息。

目录

path

Amazon S3 存储桶的文件路径。

类型：字符串

长度约束：最大长度为 256。

模式：`(s3|http|https)://.+`

必需：是

kmsKeyArn

Amazon Personalize 用来加密或解密输入和输出文件的 AWS Key Management Service (KMS) 密钥的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 2048。

模式：`arn:aws.*:kms:.*:[0-9]{12}:key/.*`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Solution

服务：Amazon Personalize

> **Important**
>
> 默认情况下，所有新解决方案都使用自动训练。如果使用自动训练，则在解决方案处于活动状态时，就会产生训练费用。为避免产生不必要的费用，在完成后，可以更新解决方案以关闭自动训练。有关训练费用的信息，请参阅 Amazon Personalize 定价。

提供有关解决方案的信息的对象。解决方案包括 Amazon Personalize 用来生成推荐的自定义配方、自定义参数和经过训练的模型（解决方案版本）。

创建解决方案后，就不能更改其配置。如果您需要进行更改，则可以使用 Amazon Personalize 控制台克隆解决方案或创建一个新的解决方案。

内容

autoMLResult

如果 `performAutoML` 为 true，请指定找到的最佳食谱。

类型：AutoMLResult 对象

必需：否

creationDateTime

解决方案的创建日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetGroupArn

提供训练数据的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

eventType

用于训练模型的事件类型（例如，“单击”或“点赞”）。如果没有提供 eventType，Amazon Personalize 将使用所有交互进行同等权重的训练，而不考虑类型。

类型：字符串

长度约束：最大长度为 256。

必需：否

lastUpdatedDateTime

上次更新解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

latestSolutionUpdate

提供解决方案最新更新的摘要。

类型：SolutionUpdateSummary 对象

必需：否

latestSolutionVersion

描述解决方案的最新版本，包括状态和 ARN。

类型：SolutionVersionSummary 对象

必需：否

name

解决方案的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

必需：否

performAutoML

> **Important**
>
> 我们建议不要启用自动机器学习。相反，请将您的使用案例与可用的 Amazon Personalize 配方匹配。有关更多信息，请参阅 Determining your use case（确定您的使用案例）。

如果为 true，Amazon Personalize 会从解决方案配置中指定的列表中搜索最佳 USER_PERSONALIZATION 食谱（不得指定 `recipeArn`）。如果为 false（默认值），Amazon Personalize 将使用 `recipeArn` 进行训练。

类型：布尔值

必需：否

performAutoTraining

指定解决方案是否自动创建解决方案版本。默认设置为 `True`，解决方案每 7 天自动创建新的解决方案版本。

有关自动训练的更多信息，请参阅创建和配置解决方案。

类型：布尔值

必需：否

performHPO

是否对所选食谱执行超参数优化（HPO）。默认值为 `false`。

类型：布尔值

必需：否

performIncrementalUpdate

一个布尔值，用于指示是否对模型执行增量训练更新。启用后，它允许模型更频繁地从新数据中学习，而无需进行全面的再训练，从而实现近乎实时的个性化。只有使用语义相似度配方的解决方案才支持此参数

类型：布尔值

必需：否

recipeArn

用于创建解决方案的食谱的 ARN。当 performAutoML 为 false 时，这是必需的。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

solutionArn

解决方案的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：arn:([a-z\d-]+):personalize:.*:.*:.+

必需：否

solutionConfig

描述解决方案的配置属性。

类型：SolutionConfig 对象

必需：否

status

解决方案的状态。

解决方案可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

SolutionConfig

服务：Amazon Personalize

描述解决方案的配置属性。

目录

algorithmHyperParameters

列出算法超参数及其值。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

autoMLConfig

包含当执行 AutoML 时要搜索的食谱列表的 AutoMLConfig 对象。

类型：AutoMLConfig 对象

必需：否

autoTrainingConfig

指定要使用的自动训练配置。

类型：AutoTrainingConfig 对象

必需：否

eventsConfig

描述事件的配置，其中包含事件参数列表。您可以指定最多 10 个事件参数。事件用于创建解决方案。

类型：EventsConfig 对象

必需：否

eventValueThreshold

仅具有大于或等于阈值的值的事件用于训练模型。

类型：字符串

长度约束：最大长度为 256。

必需：否

featureTransformationParameters

列出特征转换参数。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

hpoConfig

描述超参数优化（HPO）的属性。

类型：HPOConfig 对象

必需：否

optimizationObjective

描述解决方案的其他目标，例如最大限度地提高流时长或增加收入。有关更多信息，请参阅优化解决方案。

类型：OptimizationObjective 对象

必需：否

trainingDataConfig

指定创建自定义解决方案版本（经过训练的模型）时要使用的训练数据配置。

类型：TrainingDataConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SolutionSummary

服务：Amazon Personalize

提供解决方案属性摘要。如需完整列表，请调用 DescribeSolution API。

目录

creationDateTime

创建解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

lastUpdatedDateTime

上次更新解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

解决方案的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

recipeArn

解决方案使用的食谱的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

solutionArn

解决方案的 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

解决方案的状态。

解决方案可处于以下几种状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SolutionUpdateConfig

服务：Amazon Personalize

解决方案更新的配置详细信息。

目录

autoTrainingConfig

当 `performAutoTraining` 为 true 时要使用的自动训练配置。

类型：AutoTrainingConfig 对象

必需：否

eventsConfig

描述事件的配置，其中包含事件参数列表。您可以指定最多 10 个事件参数。事件用于创建解决方案。

类型：EventsConfig 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SolutionUpdateSummary

服务：Amazon Personalize

提供解决方案更新属性的摘要。如需完整列表，请调用 DescribeSolutionAPI。

内容

creationDateTime

解决方案更新的创建日期和时间（采用 Unix 格式）。

类型：时间戳

必需：否

failureReason

如果解决方案更新失败，则为其原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新解决方案更新的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

performAutoTraining

解决方案是否自动创建解决方案版本。

类型：布尔值

必需：否

performIncrementalUpdate

一个布尔值，用于指示是否对模型执行增量训练更新。启用后，将对具有活跃营销活动的解决方案版本进行增量训练，并允许模型更频繁地从新数据中学习，而无需进行全面的再训练，从而实现近乎实时的个性化。只有使用语义相似度配方的解决方案才支持此参数。默认值为 `true`。

请注意，某些分数和属性（例如更新物品受欢迎程度和用于通过 aws-semantic-similarity配方影响排名的新鲜度）可能要等到下一次完整训练才会更新。

类型：布尔值

必需：否

solutionUpdateConfig

解决方案的配置详细信息。

类型：SolutionUpdateConfig 对象

必需：否

status

解决方案更新的状态。解决方案更新可处于以下几种状态之一：

CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

SolutionVersion

服务：Amazon Personalize

一个对象，用于提供有关自定义数据集组中特定版解决方案的信息。

内容

creationDateTime

创建此版解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

datasetGroupArn

提供训练数据的数据集组的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

eventType

用于训练模型的事件类型（例如，“单击”或“点赞”）。

类型：字符串

长度约束：最大长度为 256。

必需：否

failureReason

如果训练解决方案版本失败，则为其原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

name

解决方案版本的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

performAutoML

如果为 true，Amazon Personalize 会根据解决方案配置搜索最佳食谱。如果为 false（默认值），Amazon Personalize 将使用 `recipeArn`。

类型：布尔值

必需：否

performHPO

是否对所选食谱执行超参数优化（HPO）。默认值为 `false`。

类型：布尔值

必需：否

performIncrementalUpdate

解决方案版本是否应执行增量更新。设置为 true 时，训练将仅处理自上次训练以来发生更改的数据，类似于 TrainingMode 设置为 UPDATE 的情况。这只能用于使用用户个性化配方的解决方案版本。

类型：布尔值

必需：否

recipeArn

解决方案中使用的食谱的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

solutionArn

解决方案的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

solutionConfig

描述解决方案的配置属性。

类型：SolutionConfig 对象

必需：否

solutionVersionArn

解决方案版本的 ARN。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

解决方案版本的状态。

解决方案版本可以处于下列状态之一：

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING
- CREATE STOPPED

类型：字符串

长度约束：最大长度为 256。

必需：否

trainingHours

用于训练模型的时间。您需要为训练模型所花费的时间付费。只有在 Amazon Personalize 成功训练模型之后，才会显示此字段。

类型：双精度

有效范围：最小值为 0。

必需：否

trainingMode

创建解决方案版本时要执行的训练的范围。FULL 训练会考虑数据集组中的所有数据。UPDATE 仅处理自上次训练以来发生变化的数据。只有通过 User-Personalization 配方创建的解决方案版本才能使用 UPDATE。

类型：字符串

有效值：`FULL | UPDATE | AUTOTRAIN`

必需：否

trainingType

解决方案版本是自动创建还是手动创建。

类型：字符串

有效值：AUTOMATIC | MANUAL

必需：否

tunedHPOParams

如果执行了超参数优化，则包含性能最佳模型的超参数值。

类型：TunedHPOParams 对象

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

SolutionVersionSummary

服务：Amazon Personalize

提供解决方案版本的属性摘要。如需完整列表，请调用 DescribeSolutionVersion API。

目录

creationDateTime

创建此版解决方案的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

failureReason

如果解决方案版本失败，则为其原因。

类型：字符串

必需：否

lastUpdatedDateTime

上次更新解决方案版本的日期和时间（采用 Unix 时间）。

类型：时间戳

必需：否

solutionVersionArn

解决方案版本的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

status

解决方案版本的状态。

解决方案版本可以处于下列状态之一：

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE - 或 - CREATE FAILED

类型：字符串

长度约束：最大长度为 256。

必需：否

trainingMode

创建解决方案版本时要执行的训练的范围。FULL 训练会考虑数据集组中的所有数据。UPDATE 仅处理自上次训练以来发生变化的数据。只有通过 User-Personalization 配方创建的解决方案版本才能使用 UPDATE。

类型：字符串

有效值：`FULL | UPDATE | AUTOTRAIN`

必需：否

trainingType

解决方案版本是自动创建还是手动创建。

类型：字符串

有效值：`AUTOMATIC | MANUAL`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Tag

服务：Amazon Personalize

您应用于资源以帮助您对其进行分类和组织的可选元数据。每个标签都包含定义的一个键和一个可选值。有关更多信息，请参阅标记 Amazon Personalize 资源。

目录

tagKey

构成标签的键-值对的一个部分。键是一种常见的标签，行为类似于更具体的标签值的类别。

类型：字符串

长度限制：长度下限为 1。最大长度为 128。

模式：`^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必需：是

tagValue

构成标签的键-值对的可选部分。值充当标签类别（键）中的描述符。

类型：字符串

长度约束：最小长度为 0。最大长度为 256。

模式：`^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ThemeGenerationConfig

服务：Amazon Personalize

使用批量推理作业生成主题的配置详细信息。

目录

fieldsForThemeGeneration

用于为批量推理作业生成描述性主题的字段。

类型：FieldsForThemeGeneration 对象

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

TrainingDataConfig

服务：Amazon Personalize

创建域推荐器或自定义解决方案版本（经过训练的模型）时要使用的训练数据配置。

内容

excludedDatasetColumns

指定要从训练中排除的列。每个键都是一个数据集类型，每个值都是一个列列表。排除列以控制 Amazon Personalize 使用哪些数据生成建议。

例如，可能有一列，您只想将其用来筛选建议。您可以将此列排除在训练之外，Amazon Personalize 只有在筛选时才会考虑该列。

类型：字符串到字符串数组映射

映射条目：最多 3 项。

密钥长度限制：最大长度为 256。

键模式：^[A-Za-z_]+$

数组成员：最多 50 项。

长度限制：最大长度为 150。

模式：[A-Za-z_][A-Za-z\d_]*

必需：否

includedDatasetColumns

指定训练中要包含的列。每个键都是一个数据集类型，每个值都是一个列列表。包括用于控制 Amazon Personalize 使用哪些数据生成推荐的列。

例如，您的项目数据集中可能有多列，但只想使用标题和描述进行训练。您只能将这两列包含在培训中，Amazon Personalize 将仅考虑培训中包含的列。

类型：字符串到字符串数组映射

映射条目：最多 3 项。

密钥长度限制：最大长度为 256。

键模式：^[A-Za-z_]+$

数组成员：最多 50 项。

长度限制：最大长度为 150。

模式：[A-Za-z_][A-Za-z\d_]*

必需：否

另请参阅

有关以特定语言之一使用此 API 的更多信息 AWSSDKs，请参阅以下内容：

- AWS适用于 C++ 的 SDK
- AWS适用于 Java 的 SDK V2
- AWS适用于 Ruby V3 的 SDK

TunedHPOParams

服务：Amazon Personalize

如果执行了超参数优化 (HPO)，则包含性能最佳模型的超参数值。

目录

algorithmHyperParameters

性能最佳模型的超参数值列表。

类型：字符串到字符串映射

映射条目：最多 100 项。

键长度限制：最大长度为 256。

值长度限制：最大长度为 1000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Amazon Personalize Events

Amazon Personalize Events 支持以下数据类型：

- Action
- ActionInteraction
- Event
- Item
- MetricAttribution
- User

Action

服务：Amazon Personalize Events

表示使用 PutActions API 添加到操作数据集的操作元数据。有关更多信息，请参阅单独导入操作。

目录

actionId

与操作关联的 ID。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

properties

操作特定的元数据的字符串映射。映射中的每个元素都由一个键值对组成。例如 {"value": "100"}。

键使用与操作数据集架构中的字段匹配的采用驼峰式大小写的名称。在前面的示例中，value 与在操作架构中定义的“VALUE”字段相匹配。对于分类字符串数据，要包含单个操作的多个类别，请使用竖线分隔符 (|) 分隔每个类别。例如 \"Deluxe|Premium\"。

类型：字符串

长度限制：最小长度为 1。最大长度为 32000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ActionInteraction

服务：Amazon Personalize Events

表示使用 PutActionInteractions API 发送的操作交互事件。

目录

actionId

用户与其交互的操作的 ID。这与操作交互架构的 ACTION_ID 字段相对应。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

eventType

操作交互事件的类型。您可以指定 Viewed、Taken 和 Not Taken 事件类型。有关操作交互事件类型数据的更多信息，请参阅事件类型数据。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

sessionId

与用户的访问关联的 ID。当用户首次访问您的网站或使用您的应用程序时，您的应用程序会生成唯一 sessionId。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

timestamp

操作交互事件发生时的时间戳。时间戳必须采用 Unix 纪元时间格式，以秒为单位。

类型：时间戳

必需：是

eventId

与事件关联的 ID。如果未提供事件 ID，则 Amazon Personalize 会为事件生成一个唯一的 ID。事件 ID 不用作模型的输入。Amazon Personalize 使用事件 ID 来区分唯一事件。模型训练中不使用第一个事件之后具有相同事件 ID 的任何后续事件。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：否

impression

操作 ID 列表，表示您已向用户展示的操作的顺序。例如 `["actionId1", "actionId2", "actionId3"]`。Amazon Personalize 不使用来自操作交互事件的展示数据。而是为每个操作记录多个事件并使用 `Viewed` 事件类型。

类型：字符串数组

数组成员：最少 1 个物品。最多 25 项。

长度限制：最小长度为 1。最大长度为 256。

必需：否

properties

您可以选择记录的事件特定数据的字符串映射。例如，如果用户采取了除操作 ID 以外的操作，您可能还会发送该用户采取的操作数量。

映射中的每个物品都由一个键值对组成。例如，

```
{"numberOfActions": "12"}
```

键使用与操作交互架构中的字段匹配的采用驼峰式大小写的名称。在上面的示例中，`numberOfActions` 将匹配在操作交互架构中定义的“NUMBER_OF_ACTIONS”字段。

以下内容不能作为属性的关键字包括在内（不区分大小写）。

- userId
- sessionId

- eventType
- timestamp
- recommendationId
- impression

类型：字符串

长度限制：长度下限为 1。最大长度为 1024。

必需：否

recommendationId

包含用户与之交互的操作的建议列表的 ID。

类型：字符串

长度限制：长度下限为 1。最大长度为 40。

必需：否

userId

与操作交互的用户的 ID。这与操作交互架构的 USER_ID 字段相对应。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Event

服务：Amazon Personalize Events

表示使用 PutEvents API 发送的物品交互事件信息。

目录

eventType

事件的类型，例如单击或下载。此属性对应于物品交互数据集架构的 EVENT_TYPE 字段，并取决于您要跟踪的事件类型。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

sentAt

事件发生时客户端的时间戳（采用 Unix 时间）。

类型：时间戳

必需：是

eventId

与事件关联的 ID。如果未提供事件 ID，则 Amazon Personalize 会为事件生成一个唯一的 ID。事件 ID 不用作模型的输入。Amazon Personalize 使用事件 ID 来区分唯一事件。模型训练中不使用第一个事件之后具有相同事件 ID 的任何后续事件。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：否

eventValue

与物品交互架构的 EVENT_VALUE 字段对应的事件值。

类型：浮点值

必需：否

impression

物品 ID 列表，代表您向用户展示的物品顺序。例如 `["itemId1", "itemId2", "itemId3"]`。提供物品列表以手动记录事件的展示数据。有关记录展示数据的更多信息，请参阅记录展示数据。

类型：字符串数组

数组成员：最少 1 个物品。最多 25 项。

长度限制：最小长度为 1。最大长度为 256。

必需：否

itemId

与物品交互数据集的架构中的 `ITEM_ID` 字段对应的物品 ID 键。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：否

metricAttribution

包含与事件关联的指标归因的相关信息。有关指标归因的更多信息，请参阅衡量建议的影响。

类型：MetricAttribution 对象

必需：否

properties

您可以选择记录的事件特定数据的字符串映射。例如，如果用户在您的网站上对电影进行评分，则除了电影 ID (`itemId`) 和评分 (`eventValue`) 外，您还可以发送该用户对电影做出的评分数量。

映射中的每个物品都由一个键值对组成。例如，

```
{"numberOfRatings": "12"}
```

键使用与物品交互数据集架构中的字段匹配的采用驼峰式大小写的名称。在上面的示例中，`numberOfRatings` 将匹配在物品交互数据集的架构中定义的“NUMBER_OF_RATINGS”字段。

以下内容不能作为属性的关键字包括在内（不区分大小写）。

- userId
- sessionId
- eventType
- timestamp
- recommendationId
- impression

类型：字符串

长度限制：长度下限为 1。最大长度为 1024。

必需：否

recommendationId

包含用户与之交互的物品的建议列表的 ID。提供 `recommendationId`，让 Amazon Personalize 隐式记录您向用户展示的建议作为展示数据。或者，在您使用指标归因来衡量建议的影响时，提供 `recommendationId`。

有关记录展示数据的更多信息，请参阅记录展示数据。有关创建指标归因的更多信息，请参阅衡量建议的影响。

类型：字符串

长度限制：长度下限为 1。最大长度为 40。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Item

服务：Amazon Personalize Events

表示使用 PutItems API 添加到物品数据集的物品元数据。有关更多信息，请参阅单独导入物品。

目录

itemId

与物品关联的 ID。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

properties

特定于物品的元数据的字符串映射。映射中的每个元素都由一个键值对组成。例如 {"numberOfRatings": "12"}。

键使用与物品数据集架构中的字段匹配的采用驼峰式大小写的名称。在前面的示例中，numberOfRatings 将匹配物品架构中定义的“NUMBER_OF_RATINGS”字段。对于分类字符串数据，要包含单个物品的多个类别，请使用竖线分隔符 (|) 分隔每个类别。例如 \"Horror|Action\"。

类型：字符串

长度限制：最小长度为 1。最大长度为 32000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

MetricAttribution

服务：Amazon Personalize Events

包含与事件关联的指标归因的相关信息。有关指标归因的更多信息，请参阅衡量建议的影响。

目录

eventAttributionSource

事件的源，例如第三方。

类型：字符串

长度限制：最大长度为 1024。

模式：`^[\x20-\x7E]*[\x21-\x7E]+[\x20-\x7E]*$`

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

User

服务：Amazon Personalize Events

表示使用 PutUsers API 添加到用户数据集的用户元数据。有关更多信息，请参阅单独导入用户。

目录

userId

与用户关联的 ID。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

必需：是

properties

用户特定元数据的字符串映射。映射中的每个元素都由一个键值对组成。例如 {"numberOfVideosWatched": "45"}。

键使用与用户数据集架构中的字段匹配的采用驼峰式大小写的名称。在前面的示例中，numberOfVideosWatched 将匹配用户架构中定义的“NUMBER_OF_VIDEOS_WATCHED”字段。对于分类字符串数据，要包含单个用户的多个类别，请使用竖线分隔符 (|) 分隔每个类别。例如 \"Member|Frequent shopper\"。

类型：字符串

长度限制：长度下限为 1。最大长度为 24000。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Amazon Personalize Runtime

Amazon Personalize Runtime 支持以下数据类型：

- PredictedAction
- PredictedItem
- Promotion

PredictedAction

服务：Amazon Personalize Runtime

标识操作的对象。

GetActionRecommendations API 返回 `PredictedAction` 的列表。

目录

actionId

建议操作的 ID。

类型：字符串

长度约束：最大长度为 256。

必需：否

score

建议操作的分数。有关操作分数的信息，请参阅操作建议评分的工作原理。

类型：双精度

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

PredictedItem

服务：Amazon Personalize Runtime

标识物品的对象。

GetRecommendations 和 GetPersonalizedRanking API 会返回 `PredictedItem` 的列表。

目录

itemId

推荐的物品 ID。

类型：字符串

长度约束：最大长度为 256。

必需：否

metadata

有关物品数据集中的物品的元数据。

类型：字符串到字符串映射

键长度限制：最大长度为 150。

值长度限制：最大长度为 20000。

必需：否

promotionName

包含预测物品的推广名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：`^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

必需：否

reason

如果您使用 User-Personalization-v2，则会有一份理由清单，说明为什么在推荐中包括该物品。可能的原因包括：

- 推广物品 - 表示您在推荐请求中应用的推广活动中包括该物品。
- 浏览 - 表示该物品包括在浏览中。使用浏览时，推荐包括的物品具有更少的交互数据或用户相关性。有关浏览的更多信息，请参阅浏览。
- 热门物品 - 表示该物品作为占位符热门物品包括在内。如果您使用筛选条件，那么根据筛选条件删除的推荐数量，Amazon Personalize 可能会添加占位符物品以满足推荐请求的 `numResults`。这些物品是基于交互数据的热门物品，符合您的筛选条件。它们没有针对用户的相关性评分。

类型：字符串数组

长度约束：最大长度为 256。

必需：否

score

模型对物品将成为下一个用户选择的确定性的数字表示。有关评分逻辑的更多信息，请参阅建议分数。

类型：双精度

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Promotion

服务：Amazon Personalize Runtime

包含有关推广的信息。推广定义了其他业务规则，这些规则适用于可配置的建议物品子集。

目录

filterArn

推广使用的筛选器的 Amazon 资源名称 (ARN)。此筛选器定义了推广物品的标准。有关更多信息，请参阅推广筛选器。

类型：字符串

长度约束：最大长度为 256。

模式：`arn:([a-z\d-]+):personalize:.*:.*:.+`

必需：否

filterValues

推广物品时要使用的值。对于推广的筛选器表达式中的每个占位符参数，提供参数名称（大小写相符）作为键，提供筛选器值作为对应的值。用英文逗号分隔一个参数的多个值。

对于使用 `INCLUDE` 元素来包含物品的筛选器表达式，您必须为表达式中定义的所有参数提供值。对于带有使用 `EXCLUDE` 元素排除物品的表达式的筛选器，您可以省略 `filter-values`。在本例中，Amazon Personalize 未使用表达式的这一部分来筛选建议。

有关创建筛选器的更多信息，请参阅筛选建议和用户细分。

类型：字符串到字符串映射

映射条目：最多 25 项。

键长度限制：最大长度为 50。

键模式：`[A-Za-z0-9_]+`

值长度限制：最大长度为 1000。

必需：否

name

推广的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 63。

模式：^[a-zA-Z0-9][a-zA-Z0-9\-_]*

必需：否

percentPromotedItems

要应用推广的推荐物品所占的百分比。

类型：整数

有效范围：最小值为 1。最大值为 100。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

常见错误类型

本节列出了此 AWS 服务可能返回的常见错误类型。并非所有服务都会返回此处列出的所有错误类型。对于特定于此服务的 API 操作的错误，请参阅该 API 操作的主题。

AccessDeniedException

您无权执行此操作。验证您的 IAM 策略是否包含所需的权限。

HTTP 状态代码：403

ExpiredTokenException

请求中包含的安全令牌已过期。申请新的安全令牌并重试。

HTTP 状态代码：403

IncompleteSignature

请求签名不符合 AWS 标准。确认您使用的是有效的 AWS 凭证，并且您的请求格式是否正确。如果您使用的是 SDK，请确保它是最新的。

HTTP 状态代码：403

InternalFailure

由于内部服务器问题，该请求现在无法处理。请稍后重试。如果问题仍然存在，请联系 Supp AWS ort。

HTTP 状态代码：500

MalformedHttpRequestException

无法处理请求正文。当无法使用指定的内容编码算法对请求正文进行解压缩时，通常会发生这种情况。验证内容编码标头是否与使用的压缩格式相匹配。

HTTP 状态代码：400

NotAuthorized

您无权执行此操作。验证您的 IAM 策略是否包含所需的权限。

HTTP 状态代码：401

OptInRequired

您的 AWS 账户需要订阅此服务。确认您已在账户中启用该服务。

HTTP 状态代码：403

RequestAbortedException

在返回响应之前，请求已中止。这通常发生在客户端关闭连接时。

HTTP 状态代码：400

RequestEntityTooLargeException

请求实体太大。请缩小请求正文的大小，然后重试。

HTTP 状态代码：413

RequestTimeoutException

请求超时。服务器未在预期的时间范围内收到完整的请求。请重试。

HTTP 状态代码：408

ServiceUnavailable

服务暂时不可用。请稍后重试。

HTTP 状态代码：503

ThrottlingException

您的请求率太高。收到此异常的 AWS SDKs 自动重试请求。请降低请求频率。

HTTP 状态代码：400

UnknownOperationException

无法识别该操作或操作。确认操作名称拼写正确，并且您使用的 API 版本是否支持该名称。

HTTP 状态代码：404

UnrecognizedClientException

您提供的 X.509 证书或 AWS 访问密钥 ID 不在我们的记录中。确认您使用的是有效的凭证，并且这些凭证尚未过期。

HTTP 状态代码：403

ValidationError

输入不符合要求的格式或限制。检查是否包含所有必需的参数以及值是否有效。

HTTP 状态代码：400

常见参数

以下列表包含所有操作用于使用查询字符串对 Signature Version 4 请求进行签名的参数。任何特定于操作的参数都列在该操作的主题中。有关签名版本 4 的更多信息，请参阅 I AM 用户指南中的签署 AWS API 请求。

X-Amz-Algorithm

您用于创建请求签名的哈希算法。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

有效值：AWS4-HMAC-SHA256

必需：条件

X-Amz-Credential

凭证范围值，该值是一个字符串，其中包含您的访问密钥、日期、您要定位的区域、您请求的服务以及终止字符串（“aws4_request”）。值采用以下格式表示：access_key/YYYYMMDD/region/service/aws4_request。

有关更多信息，请参阅 I AM 用户指南中的创建已签名的 AWS API 请求。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

X-Amz-Date

用于创建签名的日期。格式必须为 ISO 8601 基本格式 (YYYYMMDD'T'HHMMSS'Z')。例如，以下日期时间是有效 X-Amz-Date 值：20120325T120000Z。

条件： X-Amz-Date 对于所有请求都是可选的；它可用于覆盖用于签署请求的日期。如果日期标题以 ISO 8601 基本格式指定，则 X-Amz-Date 不是必填项。使用 X-Amz-Date 时，它总是会覆盖 Date 标题的值。有关更多信息，请参阅 I AM 用户指南中的 AWS API 请求签名元素。

类型：字符串

必需：条件

X-Amz-Security-Token

通过调用 AWS Security Token Service (AWS STS) 获得的临时安全令牌。有关支持来自 AWS STS的临时安全凭证的服务列表，请参阅《IAM 用户指南》中的使用 IAM 的AWS 服务。

条件：如果您使用的是中的临时安全证书 AWS STS，则必须包含安全令牌。

类型：字符串

必需：条件

X-Amz-Signature

指定从要签名的字符串和派生的签名密钥计算的十六进制编码签名。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

X-Amz-SignedHeaders

指定作为规范请求的一部分包含的所有 HTTP 标头。有关指定签名标头的更多信息，请参阅 I AM 用户指南中的创建已签名的 AWS API 请求。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

Amazon Personalize 的文档历史记录

下表描述了 Amazon Personalize 开发人员指南 每次发布时进行的重要修改。要获得本文档的更新通知，您可以订阅 RSS 源。

变更	说明	日期
新特征	Amazon Personalize 引入了语义相似度配方，该方法根据标题和描述等元数据定义项目相似度，并可按受欢迎程度或时间戳进行排名、增量培训支持以及灵活的训练栏选择。有关更多信息，请参阅语义相似度配方。	2025年11月30日
新特征	Amazon Personalize 现在支持配置解决方案，为不同的事件类型赋予不同的权重。有关更多信息，请参阅使用事件配置优化解决方案。	2025 年 4 月 11 日
新特征	Amazon Personalize 现在支持使用双栈终端节点的 IPv6 流量。有关按区域划分的 Amazon Personalize 端点的列表，请参阅《Amazon Web Services 一般参考》中的 AWS 区域和端点。	2025 年 1 月 31 日
新文档特征	Amazon Personalize 开发人员指南的结构已更新，现在包括页面浏览数据和深度分析，以提高重要主题的可见性。更新了导航以减少总体深度，并合并了相关主题。例如，您现在可以在为 Amazon Personali	2024 年 9 月 3 日

	ze 准备训练数据主题中找到所有数据准备指导。有关可以帮助您浏览新结构的主题，请参阅 Amazon Personalize 工作流程详情。本页将 Amazon Personalize 工作流程的每个步骤与指南中的相关主题对应起来。	
新特征	现在，您可以更新现有 Amazon Personalize 自定义解决方案来更改其训练配置。您可以打开或关闭自动训练，也可以更改训练频率。有关更多信息，请参阅更新解决方案以更改其自动训练配置。	2024 年 8 月 29 日
新文档特征	Amazon Personalize 开发者指南中根据 OpenSearch主题对搜索结果进行个性化设置是按 OpenSearch 服务和开源 OpenSearch 工作流程重新组织的。它现在增加了新的子主题：使用亚马逊个性化对来自亚马逊 OpenSearch 服务的结果进行个性化设置，以及使用亚马逊个性化对来自开源开放搜索的结果进行个性化设置。	2024 年 8 月 23 日
新特征	Amazon Personalize 现在支持从数据集组中删除用户及其数据，包括他们的元数据和交互数据。有关更多信息，请参阅删除用户。	2024 年 5 月 2 日

新特征	Amazon Personalize 现在可以处理多达 500 万件物品，训练速度更快、推荐延迟更短，且推荐更具相关性。有两个新的自定义配方（User-Personalization-v2 和 Personalized-Ranking-v2）使用这种增强训练。	2024 年 5 月 1 日
新特征	现在，所有新的 Amazon Personalize 自定义解决方案都使用自动训练。有关配置自动训练的更多信息，请参阅配置自动训练。	2024 年 4 月 19 日
新特征	Amazon Personalize 现在支持根据用户的行为为用户建议下一个最佳操作的功能。有关更多信息，请参阅Next-Best-Action 食谱。	2023 年 11 月 26 日
新特征	Amazon Personalize 现在支持借助生成式人工智能在批量建议中包含描述性主题的功能。有关生成具有主题的批量建议的信息，请参阅具有主题的批量建议。有关将 Amazon Personalize 与生成式人工智能结合使用的信息，请参阅 Amazon Personalize 和生成式人工智能。	2023 年 11 月 26 日

新特征	Amazon Personalize 现在支持在建议中包含物品数据集中的元数据的功能。有关为市场活动启用元数据的信息，请参阅推荐中的物品元数据。有关为推荐器启用元数据的信息，请参阅在推荐器建议中启用元数据。	2023 年 11 月 26 日
新特征	Amazon Personalize 现在支持通过 OpenSearch 服务对搜索结果进行个性化设置。有关更多信息，请参阅对搜索结果进行个性化设置。 OpenSearch	2023 年 10 月 16 日
新特征	Amazon Personalize 现在支持以下功能：为物品数据集导入最多 100 个元数据列（原为 50 个），为用户数据集导入最多 25 个元数据列（原为 5 个）。有关 Amazon Personalize 限制的更多信息，请参阅 Amazon Personalize 端点和配额。	2023 年 9 月 5 日
新特征	Amazon Personalize 现在支持对来自的搜索结果进行个性化设置 OpenSearch。有关更多信息，请参阅个性化来自 OpenSearch （自行管理）的搜索结果。	2023 年 7 月 25 日
新特征	Amazon Personalize 现在支持以下功能：将数据集的架构替换为新的或现有架构。有关更多信息，请参阅替换数据集的架构。	2023 年 7 月 13 日

新特征	如果您使用 User-Personalization 或 Personalized-Ranking 食谱，则 Amazon Personalize 批量推理作业现在可以使用您以增量方式导入的数据，而无需重新训练。有关更多信息，请参阅获取批量建议。	2023 年 6 月 30 日
新特征	Amazon Personalize 现在支持以下功能：根据您在请求相关物品建议时指定的物品筛选物品。有关筛选器的信息，请参阅筛选建议和用户细分。	2023 年 6 月 21 日
新特征	Amazon Personalize 现在支持以下功能：根据您在请求相关物品建议时指定的物品筛选物品。有关筛选器的信息，请参阅筛选建议和用户细分。	2023 年 6 月 21 日
新特征	Amazon Personalize 现在支持通过接口 Amazon VPC 端点在虚拟私有云 (VPC) 和 Amazon Personalize 之间建立私有连接。有关更多信息，请参阅 Amazon Personalize 和接口 VPC 端点 (AWS PrivateLink)。	2023 年 6 月 12 日
新特征	Amazon Personalize 现在支持在创建推荐器或自定义解决方案时配置训练时使用的列。有关在创建推荐器时配置列的信息，请参阅创建推荐器。有关在创建解决方案时配置列的信息，请参阅配置训练时使用的列。	2023 年 5 月 30 日

新文档特征	Amazon Personalize 开发者指南现在包含有关使用亚马逊个性化建议进行 A/B 测试的信息。有关更多信息，请参阅通过 A/B 测试衡量推荐的影响。	2023 年 5 月 5 日
新特征	Amazon Personalize 现在支持配置受欢迎程度如何影响 Similar-Items 食谱生成的建议。有关更多信息，请参阅类似物品食谱。	2023 年 4 月 21 日
新特征	Amazon Personalize 现在支持使用亚马逊 SageMaker AI Data Wrangler 将来自 40 多个来源的数据导入亚马逊个性化数据集。有关更多信息，请参阅使用 Amazon A SageMaker I Data Wrangler 导入数据。	2023 年 4 月 14 日
新文档特征	Amazon Personalize 开发人员指南现在包括一份新的准备清单。此清单可帮助您准备使用 Amazon Personalize 处理自己的数据。有关更多信息，请参阅准备清单。	2023 年 2 月 9 日
新特征	Amazon Personalize 现在支持为您导入数据集的数据生成见解和统计信息。有关更多信息，请参阅分析数据集中的数据。	2023 年 1 月 25 日

新特征	Amazon Personalize 现在支持自定义数据集组的全新 Trending-Now 食谱。有关更多信息，请参阅 Trending-Now 食谱。	2023 年 1 月 6 日
新特征	Amazon Personalize 现在支持在 IAM 策略中使用标签。有关更多信息，请参阅在 IAM 策略中使用标签。	2022 年 12 月 28 日
新特征	Amazon Personalize 现在支持调整模型在训练期间考虑的最大交互数 配额。此外，模型在训练期间考虑的最大用户数限制 配额不再适用。有关更多信息，请参阅 Amazon Personalize 端点和配额。	2022 年 12 月 15 日
新特征	Amazon Personalize 现在支持创建指标归因来衡量建议对业务的影响。有关更多信息，请参阅衡量建议的影响。	2022 年 11 月 17 日
新特征	Amazon Personalize 的现用解决方案、现用市场活动、推荐器和筛选器总数的配额现已增加。现在，其中每个配额都适用于每个数据集组，而不是每个账户。有关配额的信息，请参阅 Amazon Personalize 端点和配额。	2022 年 9 月 7 日

新特征	Amazon Personalize 筛选器现在考虑每位用户每种事件类型最多 100 次交互。有关筛选器的信息，请参阅筛选建议和用户细分。	2022 年 8 月 29 日
新特征	Amazon Personalize 现在支持 VIDEO_ON_DEMAND 域的新当前热门使用案例。有关更多信息，请参阅 VIDEO_ON_DEMAND 使用案例。	2022 年 8 月 17 日
新特征	Amazon Personalize 现在支持使用单独的推广筛选器在建议中推广物品。有关推广物品的信息，请参阅在建议中推广物品。	2022 年 8 月 12 日
新特征	Amazon Personalize 现在支持在带占位符参数的筛选器表达式中使用比较运算符。有关筛选器表达式的信息，请参阅筛选器表达式。	2022 年 8 月 12 日
新特征	Amazon Personalize 现在支持对数据集进行增量批量更新。现在，您可以使用数据集导入作业来更新数据集，而无需替换现有数据。有关更多信息，请参阅更新现有批量数据。	2022 年 8 月 2 日
新特征	Amazon Personalize 现在可以使用不同语言的非结构化文本元数据。有关更多信息，请参阅非结构化文本元数据。	2022 年 6 月 6 日

新特征	Amazon Personalize 推荐器现在可以生成离线指标。您可使用这些指标来评估您的推荐器的性能。有关更多信息，请参阅评估推荐器。	2022 年 5 月 24 日
新特征	Amazon Personalize 现在支持停止推荐器然后稍后重启的功能。这样，您就可以暂停推荐器计费，并仅在使用推荐器时才付费。有关更多信息，请参阅停止和启动推荐器。	2022 年 4 月 20 日
新特征	Amazon Personalize 现在支持使用标签对 Amazon Personalize 资源进行分类和管理。有关更多信息，请参阅标记 Amazon Personalize 资源。	2022 年 4 月 7 日
新特征	Amazon Personalize 现在支持使用指定资源CloudFormation。有关更多信息，请参阅使用指定资源CloudFormation。	2022 年 3 月 11 日
新文档特征	Amazon Personalize 开发人员指南现在包括一个新的问题排查主题，该主题提供常见问题解答以及您在使用 Amazon Personalize 时可能遇到的错误消息的问题排查建议。有关更多信息，请参阅故障排除。	2022 年 2 月 15 日

新特征	Amazon Personalize 现在支持使用针对点播视频或电子商务域的使用案例优化资源创建域数据集组。有关更多信息，请参阅域数据集组。	2021 年 11 月 29 日
新特征	Amazon Personalize 现在支持使用新的 USER_SEGMENTATION 食谱创建用户细分。USER_SEGMENTATION 食谱根据物品输入数据生成用户细分。有关更多信息，请参阅 Item-Affinity 配方和配方。Item-Attribute-Affinity	2021 年 11 月 29 日
新特征	Amazon Personalize 现在支持新的 RELATED_ITEMS 建议食谱 Similar-Items。使用 Similar-Items 食谱，可根据交互数据和物品元数据为相似物品生成建议。有关更多信息，请参阅类似物品食谱。	2021 年 10 月 5 日
新文档特征	Amazon Personalize 开发人员指南现在包括一个入门教程，介绍如何结合使用 Amazon Personalize 与 SDK for Java 2.x。有关更多信息，请参阅使用入门 (SDK for Java 2.x)。	2021 年 8 月 25 日
新特征	Amazon Personalize 现在可以从物品数据集的非结构化文本元数据中提取有意义的信息。有关更多信息，请参阅物品数据集。	2021 年 6 月 9 日

新特征	Amazon Personalize 现在支持停止创建解决方案版本（停止训练模型）的功能。有关更多信息，请参阅停止创建解决方案版本。	2021 年 5 月 20 日
新特征（预览版）	除了最大限度地提高相关性（例如最大限度地提高收入）之外，Amazon Personalize 现在还可以针对目标优化解决方案。此特征提供预览版。有关更多信息，请参阅针对其他目标优化解决方案。	2021 年 5 月 18 日
新特征	Amazon Personalize 现在可以将 Amazon Personalize 数据集中的记录导出到 Amazon S3 存储桶中进行分析和跟踪。有关更多信息，请参阅导出数据集。	2021 年 4 月 26 日
新特征	现在，Amazon Personalize 会每两小时自动更新一次您使用 User-Personalization 训练的最新模型（解决方案版本），以包含新数据。有关更多信息，请参阅 User-Personalization 食谱。	2020 年 11 月 17 日
新特征	Amazon Personalize 现在可以根据您在获得建议时指定的标准筛选建议。有关更多信息，请参阅筛选建议。	2020 年 11 月 10 日

新特征	Amazon Personalize 现在支持以增量方式导入用户和物品的功能。有关更多信息，请参阅以增量方式导入记录。	2020 年 10 月 2 日
新特征	Amazon Personalize 现在支持新的 USER_PERSONALIZATION 建议食谱。USER_PERSONALIZATION 特征包括建模展示数据、自动物品浏览和自动冷物品选择。有关更多信息，请参阅 User-Personalization 食谱。	2020 年 8 月 5 日
新特征	Amazon Personalize 现在可以使用自定义筛选器表达式根据物品和用户元数据筛选建议。有关更多信息，请参阅筛选建议。	2020 年 7 月 31 日
新特征	Amazon Personalize 现在允许您根据用户与哪些物品交互来筛选结果。有关更多信息，请参阅筛选建议。	2020 年 6 月 3 日
新特征	Amazon Personalize 现在公开建议物品的分数。分数表示 Amazon Personalize 模型对于用户接下来将选择某个物品的确定性。有关更多信息，请参阅获取建议。	2020 年 4 月 3 日

新区域	Amazon Personalize 增加了对亚太地区（首尔）区域的支持。有关 Amazon Personalize 支持的 AWS 区域的完整列表，请参阅《Amazon Web Services 一般参考》中的 AWS 区域表或 AWS 区域和端点。	2020 年 1 月 21 日
新特征	Amazon Personalize 现在可根据上下文元数据获取建议。有关更多信息，请参阅获取建议。	2019 年 12 月 19 日
新区域	Amazon Personalize 增加了对亚太地区（孟买）、亚太地区（悉尼）和加拿大（中部）区域的支持。有关 Amazon Personalize 支持的AWSAWS区域的完整列表，请参阅亚马逊网络服务一般参考中的区域表或AWS地区和终端节点。	2019 年 12 月 18 日
新特征	Amazon Personalize 现在支持批量建议工作流程。有关更多信息，请参阅获取批量建议。	2019 年 11 月 14 日
Amazon Personalize 通用版	Amazon Personalize 现可用于一般用途。	2019 年 6 月 10 日
Amazon Personalize 预览版	这是 Amazon Personalize 文档的第一个预览版本。	2018 年 11 月 28 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。