



Movies Recommender by Personalize

Hands on Lab

March 2019

Overview

Amazon Personalize is a machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications.

With Amazon Personalize, you provide an activity stream from your application – page views, signups, purchases, and so forth – as well as an inventory of the items you want to recommend, such as articles, products, videos, or music. You can also choose to provide Amazon Personalize with additional demographic information from your users such as age, or geographic location. Amazon Personalize will process and examine the data, identify what is meaningful, select the right algorithms, and train and optimize a personalization model that is customized for your data.

Movies Recommender with Personalize

In this lab you will build a movies recommender with Amazon Personalize. The problem statement is, given a historical rating of movies by users, the system should be able to recommend movies that a particular user may like.

Amazon Personalize, as a managed recommendation engine service by AWS, will be used to create the data schema, build data set, import data, train the engine, deploy the solution, and host the recommendation inference.

Prepare data

Please set-up an AWS SageMaker Notebook Instance, open Jupyter Notebook, and complete Step PS1 until Step PS3 in https://github.com/yudho/machine-learning-workshop/blob/master/recommendation-engine-with-personalize/personalize_sample_notebook.ipynb

Prepare S3 bucket access

1. Open S3 <https://s3.console.aws.amazon.com/s3/home?region=us-east-1>
2. Take note of the S3 bucket name that ends with “-product-recommendation-personalize”. We will use it later.
3. Click the S3 bucket name
4. Go to “Permissions” tab, and click “Bucket Policy”.
5. Paste this policy into the Bucket policy editor, change the bolded “<s3 bucket name>” to the name of this S3 bucket and click Save:

```
{
  "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:::<s3 bucket name>" ,  
"arn:aws:s3:::<s3 bucket name>/*"  
  
]  
  
}  
  
]  
  
}
```

Create dataset group, dataset, schema, and import job

1. Open Amazon Personalize console in N.Virginia region by following <https://console.aws.amazon.com/personalize/home?region=us-east-1>
2. If it displays a homepage (since you do not have any Amazon Personalize resources in that region), then click "Get Started"
3. The Create dataset group page should be displayed
4. Enter this detail and click Next
 - a. Dataset group name: **movie100-dataset-group**
 - b. Input data: **upload user-item interaction data**
5. This page below will be displayed. Enter detail:
 - a. Dataset name: **ratings**
 - b. Schema selection: **Create new schema**
 - c. New schema name: **movie100-schema**

Dataset details

Dataset name

The text you type here appears in the Dataset dashboard and detail page. It can help you distinguish this dataset from others.

The dataset name must have 2 to 256 characters. Valid characters: a-z, A-Z, 0-9, and . : + = @ % - (hyphen).

Schema details

Schema selection [Info](#)



Use existing schema

Choose an existing schema that matches your dataset.



Create new schema

Create a new schema to match your dataset

New schema name

The text you type here appears in the Schema dashboard. It can help you distinguish this schema from others.

The schema name must have 2 to 256 characters. Valid characters: a-z, A-Z, 0-9, and . : + = @ % - (hyphen).

6. For Schema definition, enter this JSON document below. Then click Next

```
{
  "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"  
}
```

7. The Import user-item interaction data should be displayed. Otherwise, an error might happen. In that case just wait for a while and repeat step 1-6, except that, instead of creating new schema, select an existing one.
8. Fill in this detail and click Next:
 - a. Dataset import job name: **movie100-dataset-import-job**
 - b. Automatic dataset import jobs: **Off**
 - c. IAM service role:
 - i. click 'Create a new role' and when prompted, select 'Specific S3 buckets' under 'S3 buckets you specify' dialog, and for the value, paste the bucket name that you have taken note in step 2 in "Prepare S3 bucket access" section above
 - d. Data location: **s3://<bucket_name>/DEMO-movie-lens-100k.csv**
 - i. Please substitute <bucket_name> with the name of your S3 bucket from step 2 in "Prepare S3 bucket access" section above
9. You will be directed to the movie100-dataset-group Dashboard page. Please wait until dataset finish importing

Amazon Personalize ×

Amazon Personalize > Dataset groups > movie100-dataset-group > Dashboard

Dashboard

Overview

Upload datasets

Datasets are required to create solutions, which are then used to generate recommendations.

User-item interaction data [Create in progress](#) [view](#)

User data [Import](#)

Install event ingestion SDK

The event ingestion SDK allows you to track user events in your application and feed them to your solutions.

SDK Installation [Start](#)

Create solutions

Solutions help you generate recommendations. They consist of custom models trained on your datasets along with the underlying infrastructure required to generate recommendations.

Solution training [Start](#)

Launch campaigns

Campaigns create an endpoint that your application calls to get recommendations from a solution. They also provide you with analytics on the solution's usage.

Campaign creation [Start](#)

Create solution

After we have all the data-related stuff, now we need to train the recommender engine with those ratings data we have. In Amazon Personalize, creating solution means training the engine.

In this step, we can choose either to train with specific recipe/algorithm (e.g. HRNN), or do AutoML and let the Personalize organize the training with multiple recipes/algorithms and choose the best one.

1. On movie100-dataset-group Dashboard page, click “Start” on Solution training
2. For Solution name, enter **movie100-solution**
3. For Recipe selection, choose **Automatic (AutoML)**.
4. For Event type and Minimum transactions per second, leave the default values.
5. Click Finish

[Amazon Personalize](#) > [Dataset groups](#) > [movie100-dataset-group](#) > [Solutions](#) > [Create solution](#)

Create solution [Info](#)

You create a solution using a recipe that is tailored to a specific use case.

Solution configuration

Solution name

The solution name that you type here can help you distinguish this solution from other solutions.

The solution name must have 2 to 256 characters. Valid characters: a-z, A-Z, 0-9, and . : + = @ % - (hyphen).

Minimum provisioned transactions per second [Info](#)

The minimum amount of throughput in transactions per second (TPS) that is provisioned for this solution.

Type in a value between 1 and 500.

Recipe selection [Info](#)

A recipe consists of recommendation algorithms and data processing steps needed to create a solution. Amazon Personalize has several prebuilt recipes.

☐ **Manual**

Choose the recipe manually.

☒ **Automatic (AutoML)**

Amazon Personalize will find the best recipe for your dataset.

AutoML recipe list

Select candidate recipes that you would like Amazon Personalize to examine and determine the best recipe. By default, Amazon Personalize will train on DeepFM, FFNN, and HRNN recipes when performing AutoML.

AutoML recipe list

Select candidate recipes that you would like Amazon Personalize to examine and determine the best recipe. By default, Amazon Personalize will train on DeepFM, FFNN and HRNN recipes when performing AutoML.

Select a recipe ▼

aws-deepfm

Predicts items a user will interact with. Extends Factorization machine using a deep autoencoder to achieve higher accuracy via nonlinear modeling of user-item interactions.

aws-ffnn

Predicts items a user will interact with. A feed-forward neural network that utilizes a built-in K-nearest neighbors search to deliver recommendations based on past interactions in the user-item interactions dataset.

aws-hrnn

Predicts items a user will interact with. A Hierarchical Recurrent Neural Network which models the temporal order of user-item interactions.

Launch campaign

After we have trained the engine, we should deploy our recommender engine to a server/container. In Amazon Personalize, we can launch campaign to actually deploy the trained solution.

1. On movie100-dataset-group Dashboard page, click 'Start' on Campaign creation
2. For Campaign name, enter **movie100-campaign**
3. For Solution, select the solution that we created in the section 'Create solution' above
4. Click Create campaign

Get recommendation

Now, it is time to test our recommendation engine by making inference.

1. Go to <https://docs.aws.amazon.com/personalize/latest/dg/getting-started-console.html>
2. Expand the section 'Step 4: Get Recommendations'
3. Follow the instructions in there to test the recommendation
4. Good, we see some IDs, but what movies are they? Let's follow the steps below to get recommendation from SageMaker Notebook Instance where we can define the movie titles mappings
5. From AWS Personalize console, locate the campaign under movie100-dataset-group.
6. Identify the ARN of the campaign and note it down
7. Go back to the SageMaker Notebook that we used earlier in this lab and go straight to Step PS11.

8. Run 'Select a user' step
9. Under 'Call GetRecomendations' section, change the campaignArn value to the ARN you noted in step 6 above, enclosed in single quote to make it a string
10. Run that section
11. You should see the list of recommended movies for that user