**Train a simple recommender**

The main aim of a recommendation system is to recommend one or more items to users of the system. Examples of an item to be recommended, might be a movie, restaurant, book, or song. In general, the user is an entity with item preferences such as a person, a group of persons, or any other type of entity you can imagine.

There are two principal approaches to recommender systems:

* The content-based approach, which makes use of features for both users and items. Users can be described by properties such as age or gender. Items can be described by properties such as the author or the manufacturer. Typical examples of content-based recommendation systems can be found on social matchmaking sites.
* The Collaborative filtering approach, which uses only identifiers of the users and the items. It is based on a matrix of ratings given by the users to the items. The main source of information about a user is the list the items they’ve rated and the similarity with other users who have rated the same items.

The SVD recommender module in Azure Machine Learning designer is based on the Single Value Decomposition algorithm. It uses identifiers of the users and the items, and a matrix of ratings given by the users to the items. It’s a typical example of collaborative recommender.

Lab Overview

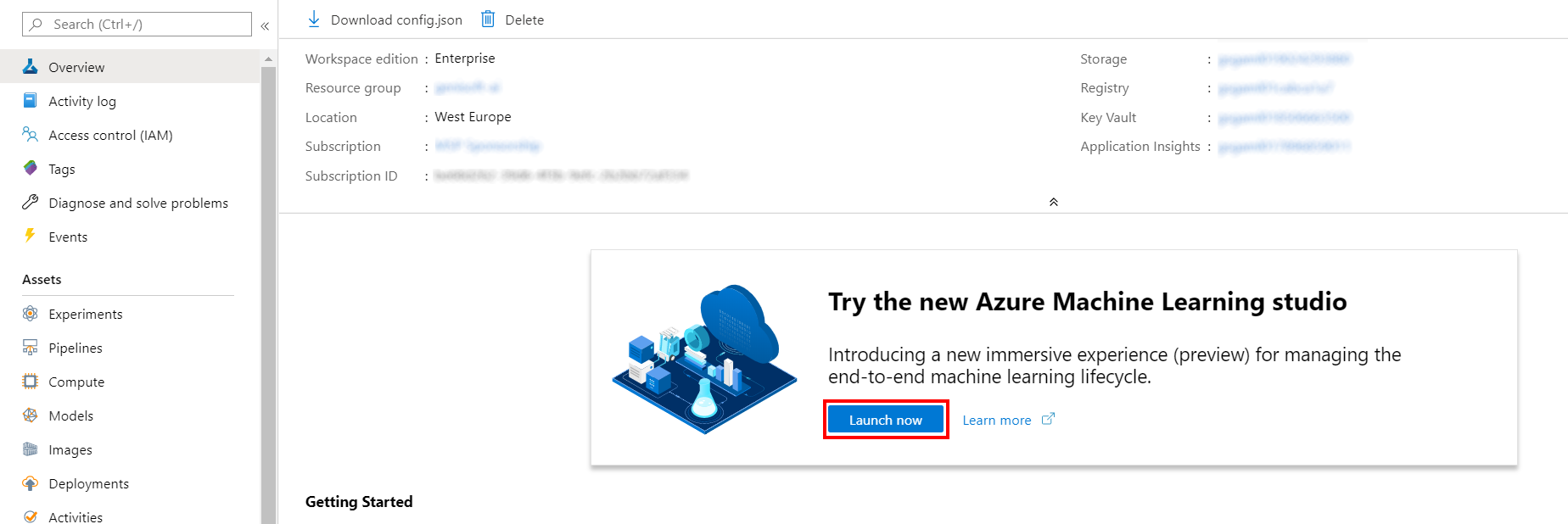
In this lab, we make use of the Train SVD Recommender module available in Azure Machine Learning designer (preview), to train a movie recommender engine. We use the collaborative filtering approach: the model learns from a collection of ratings made by users on a subset of a catalog of movies. Two open datasets available in Azure Machine Learning designer are used the IMDB Movie Titles dataset joined on the movie identifier with the Movie Ratings dataset. The Movie Ratings data consists of approximately 225,000 ratings for 15,742 movies by 26,770 users, extracted from Twitter using techniques described in the original paper by Dooms, De Pessemier and Martens. The paper and data can be found on [GitHub](https://github.com/sidooms/MovieTweetings).

We will both train the engine and score new data, to demonstrate the different modes in which a recommender can be used and evaluated. The trained model will predict what rating a user will give to unseen movies, so we’ll be able to recommend movies that the user is most likely to enjoy. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

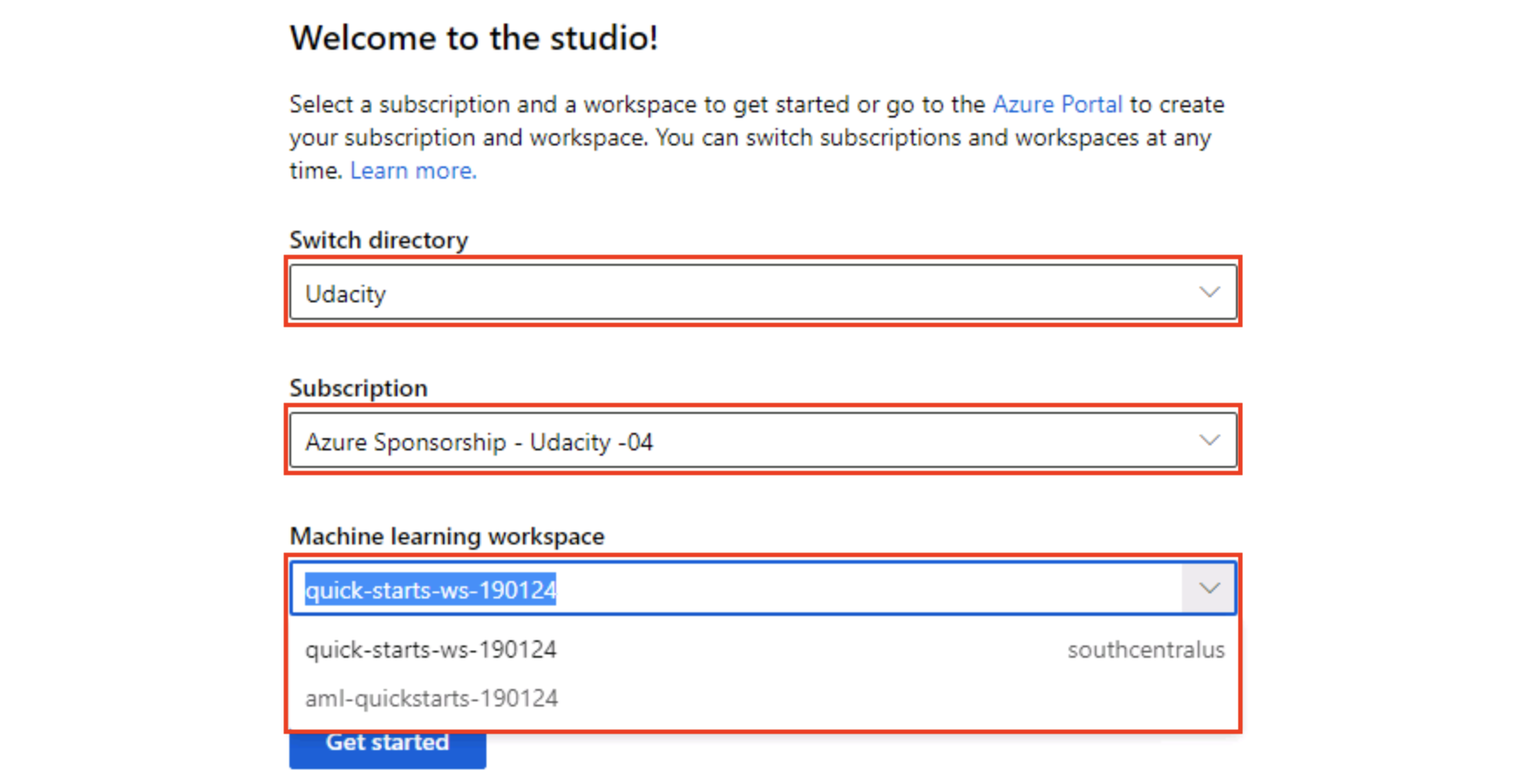
Exercise 1: Create New Training Pipeline

Task 1: Open Pipeline Authoring Editor

1. In [Azure portal](https://portal.azure.com/), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.

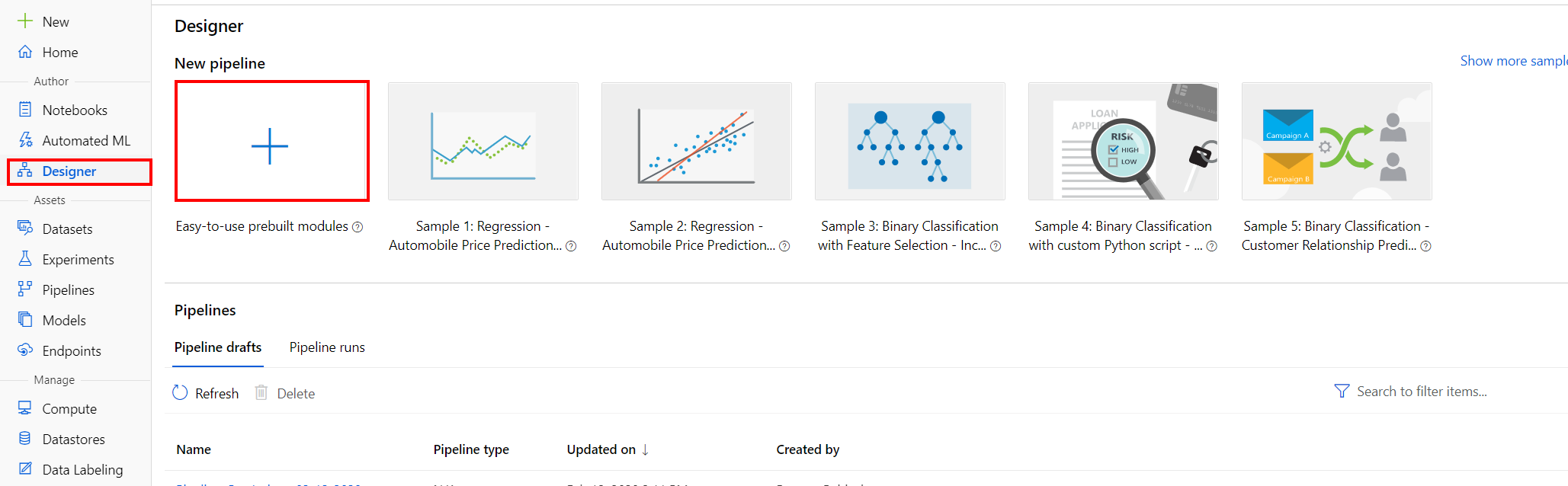


1. When you first launch the studio, you may need to set the directory and subscription. If so, you will see this screen:



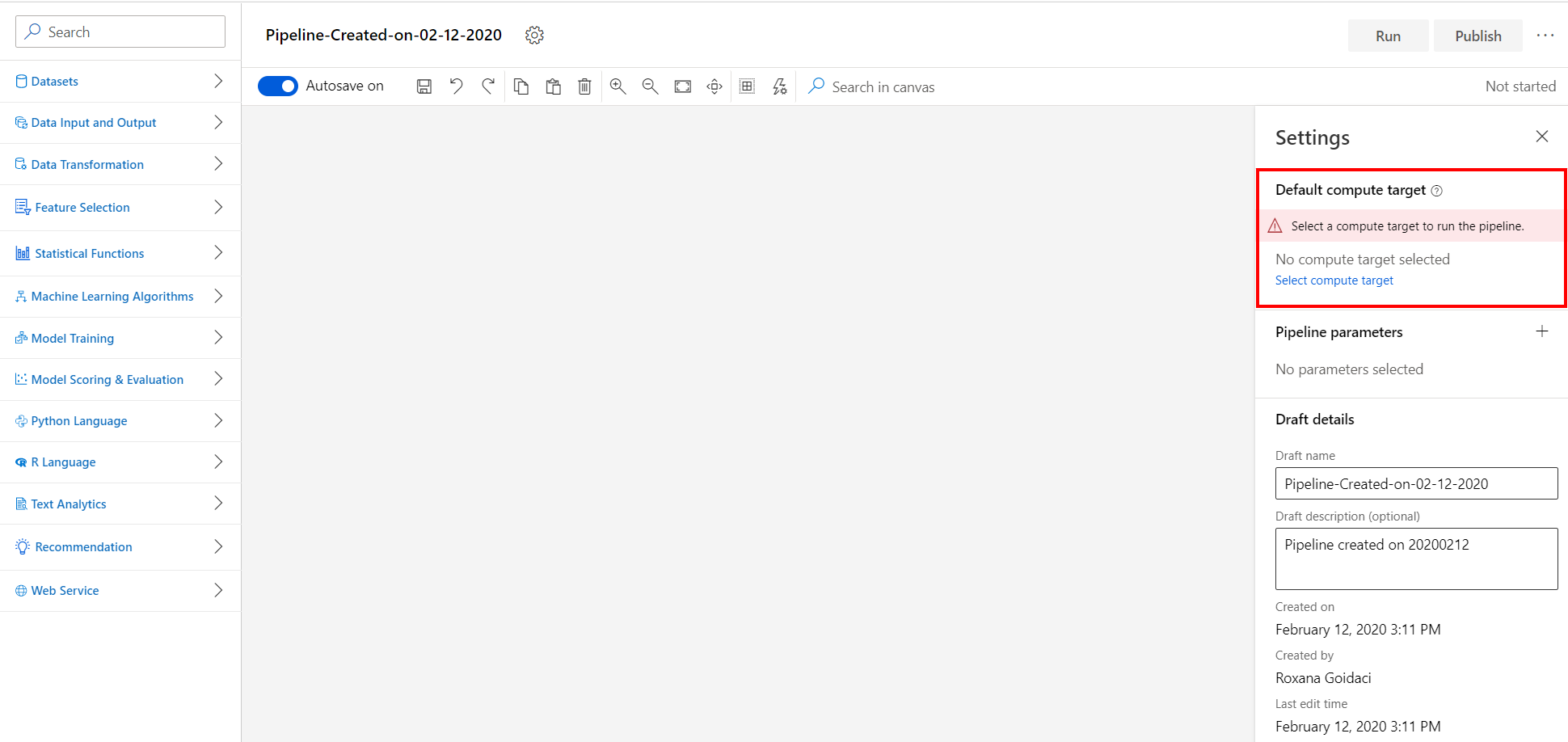
For the directory, select **Udacity** and for the subscription, select **Azure Sponsorship**. For the machine learning workspace, you may see multiple options listed. **Select any of these** (it doesn’t matter which) and then click **Get started**.

1. From the studio, select **Designer, +**. This will open a visual pipeline authoring editor.



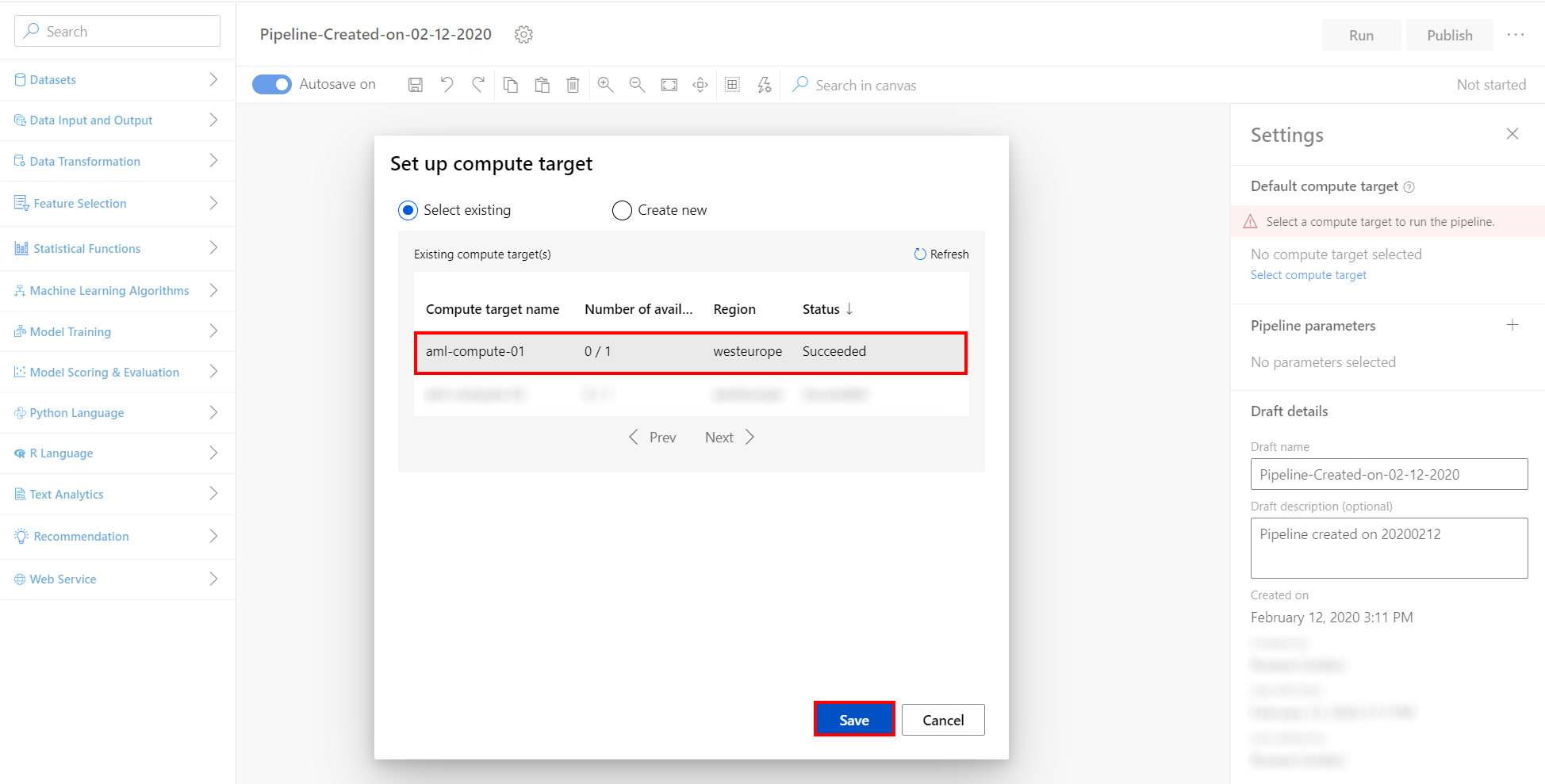
Task 2: Setup Compute Target

1. In the settings panel on the right, select **Select compute target**.



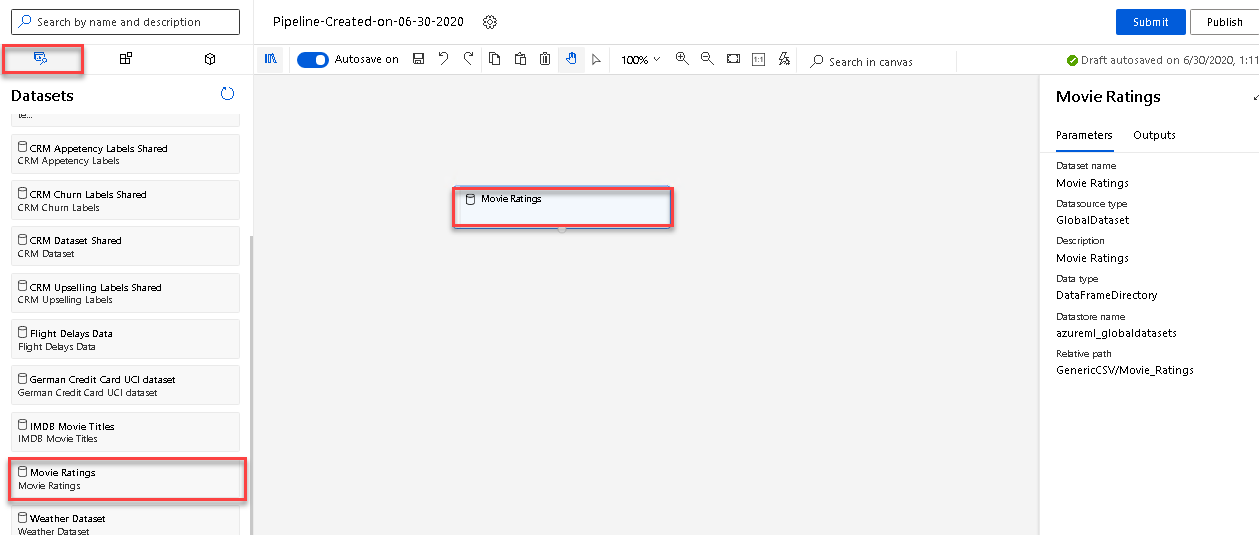
1. In the Set up compute target editor, select the available compute, and then select **Save**.

Note: If you are facing difficulties in accessing pop-up windows or buttons in the user interface, please refer to the Help section in the lab environment.

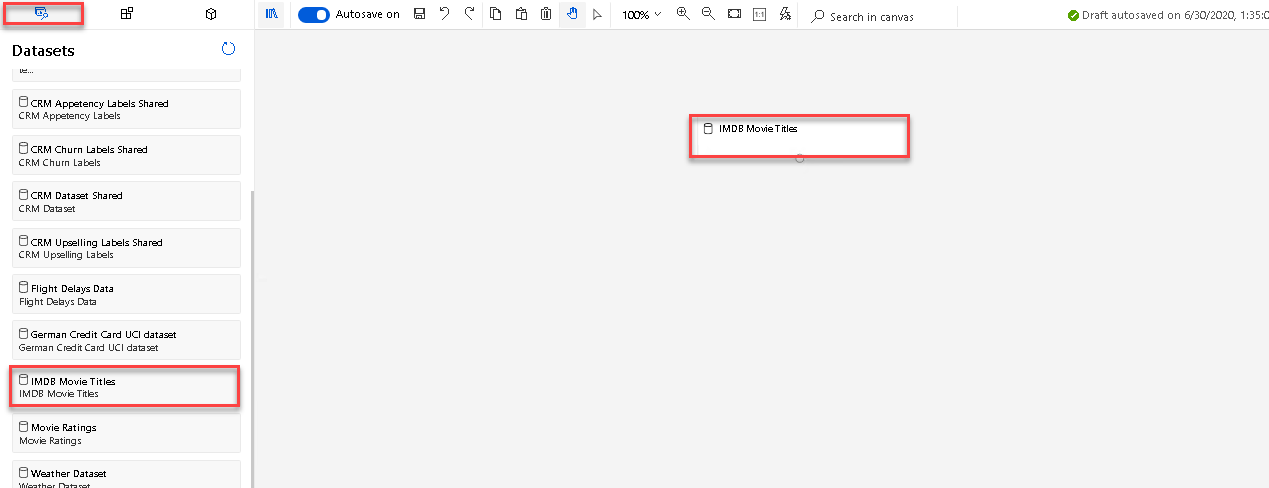


Task 3: Add Sample Datasets

1. Select **Datasets** section in the left navigation. Next, select **Samples, Movie Ratings** and drag and drop the selected dataset on to the canvas.

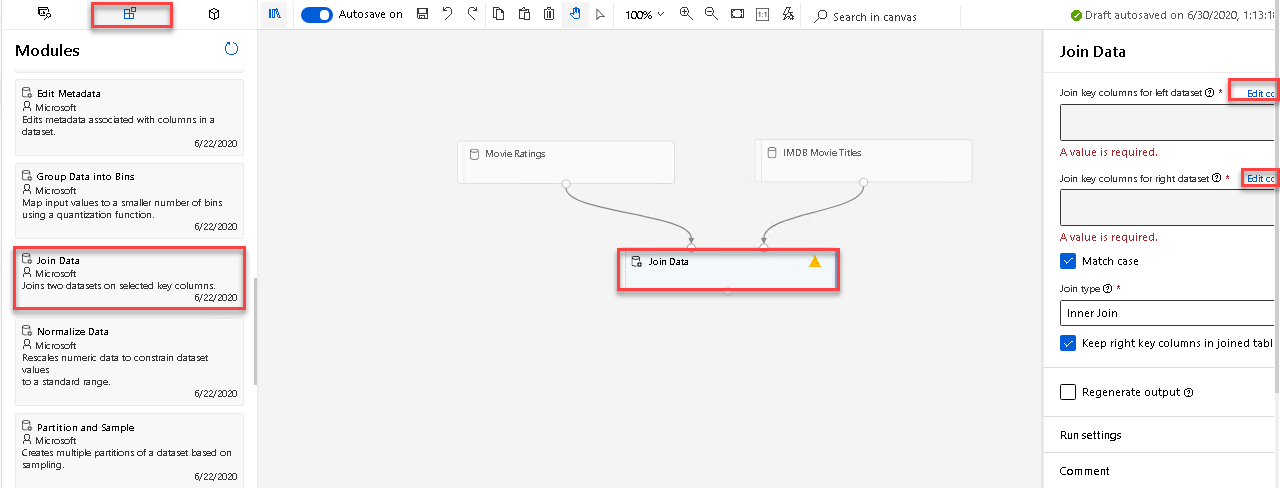


1. Select **Datasets** section in the left navigation. Next, select **Samples, IMDB Movie Titles** and drag and drop the selected dataset on to the canvas.



Task 4: Join the two datasets on Movie ID

1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Join Data** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the output of the Movie Ratings module to the first input of the Join Data module.
   4. Connect the output of the IMDB Movie Titles module to the second input of the Join Data module.



1. Select the Join Data module.
2. Select the **Edit column** link to open the Join key columns for left dataset editor. Select the **MovieId** column in the Enter column name field.

![Image shows the Join key columns for left dataset` editor.](images/07.png ‘Join key columns for left dataset’)

1. Select the **Edit column** link to open the Join key columns for right dataset editor. Select the **Movie ID** column in the Enter column name field.

![Image shows the Join key columns for left dataset` editor.](images/08.png ‘Join key columns for left dataset’)

*Note that you can submit the pipeline at any point to peek at the outputs and activities. Running pipeline also generates metadata that is available for downstream activities such selecting column names from a list in selection dialogs.*

Task 5: Select Columns UserId, Movie Name, Rating using a Python script

1. Select **Python Language** section in the left navigation. Follow the steps outlined below:
   1. Select the **Execute Python Script** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the Join Data output to the input of the Execute Python Script module.

![Image shows the Select Columns in Dataset module added to the canvas` editor.](images/09.png ‘Add Select Columns in Dataset module’)

1. Select **Edit code** to open the Python script editor, clear the existing code and then enter the following lines of code to select the UserId, Movie Name, Rating columns from the joined dataset. Please ensure that there is no indentation for the first line and the second and third lines are indented.
2. def azureml\_main(dataframe1 = None, dataframe2 = None):
3. df1 = dataframe1[['UserId','Movie Name','Rating']]

return df1,

Note: In other pipelines, for selecting a list of columns from a dataset, we could have used the Select Columns from Dataset prebuilt module. This one returns the columns in the same order as in the input dataset. This time we need the output dataset to be in the format: user id, movie name, rating.This column order is required at the input of the Train SVD Recommender module.

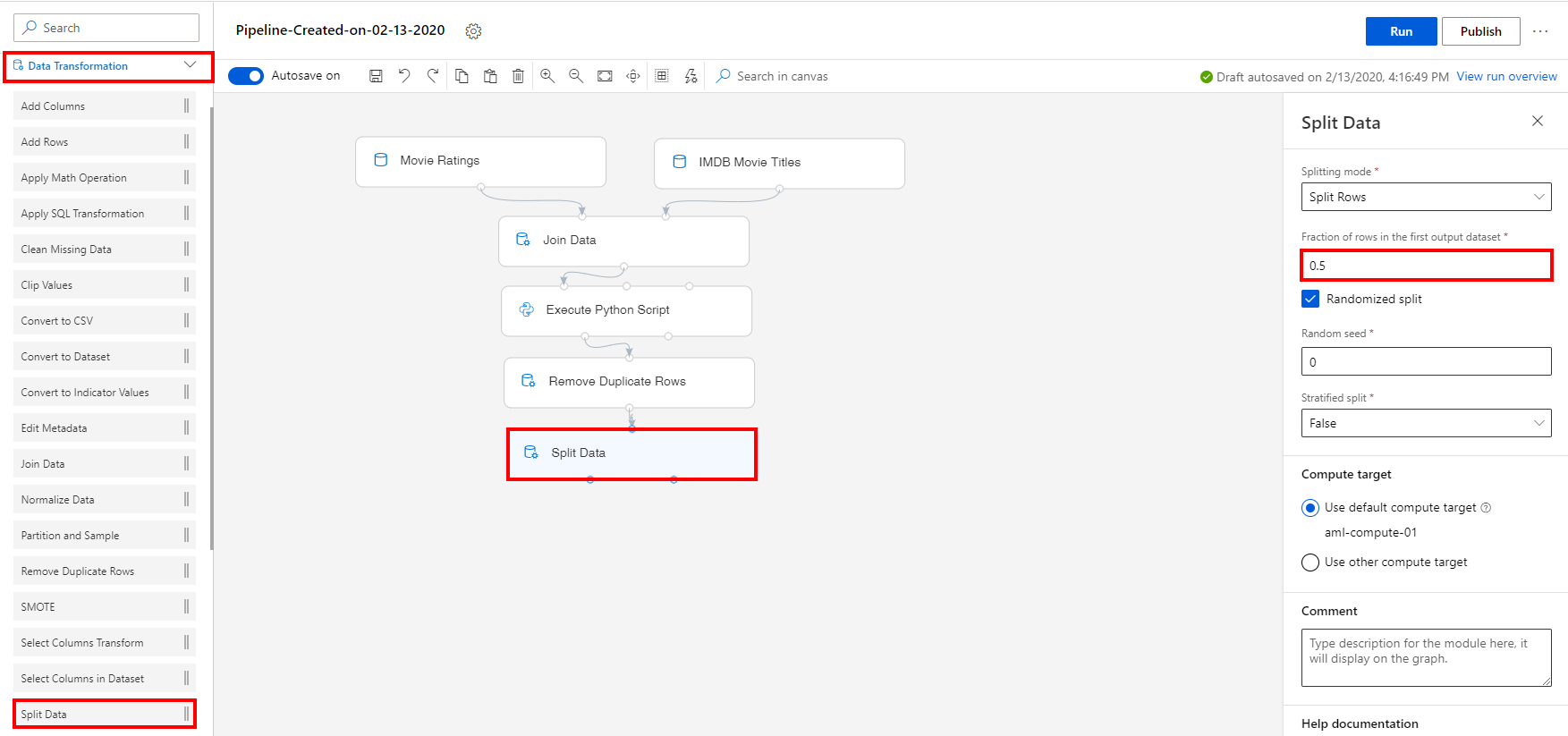
Task 6: Remove duplicate rows with same Movie Name and UserId

1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Remove Duplicate Rows** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the first output of the Execute Python Script to the input of the Remove Duplicate Rows module.
   4. Select the **Edit columns** link to open the Select columns editor and then enter the following list of columns to be included in the output dataset: **Movie Name**, **UserId**.

![Image shows the Remove Duplicate Rows module added to the canvas` editor.](images/10.png ‘Add Remove Duplicate Rows module’)

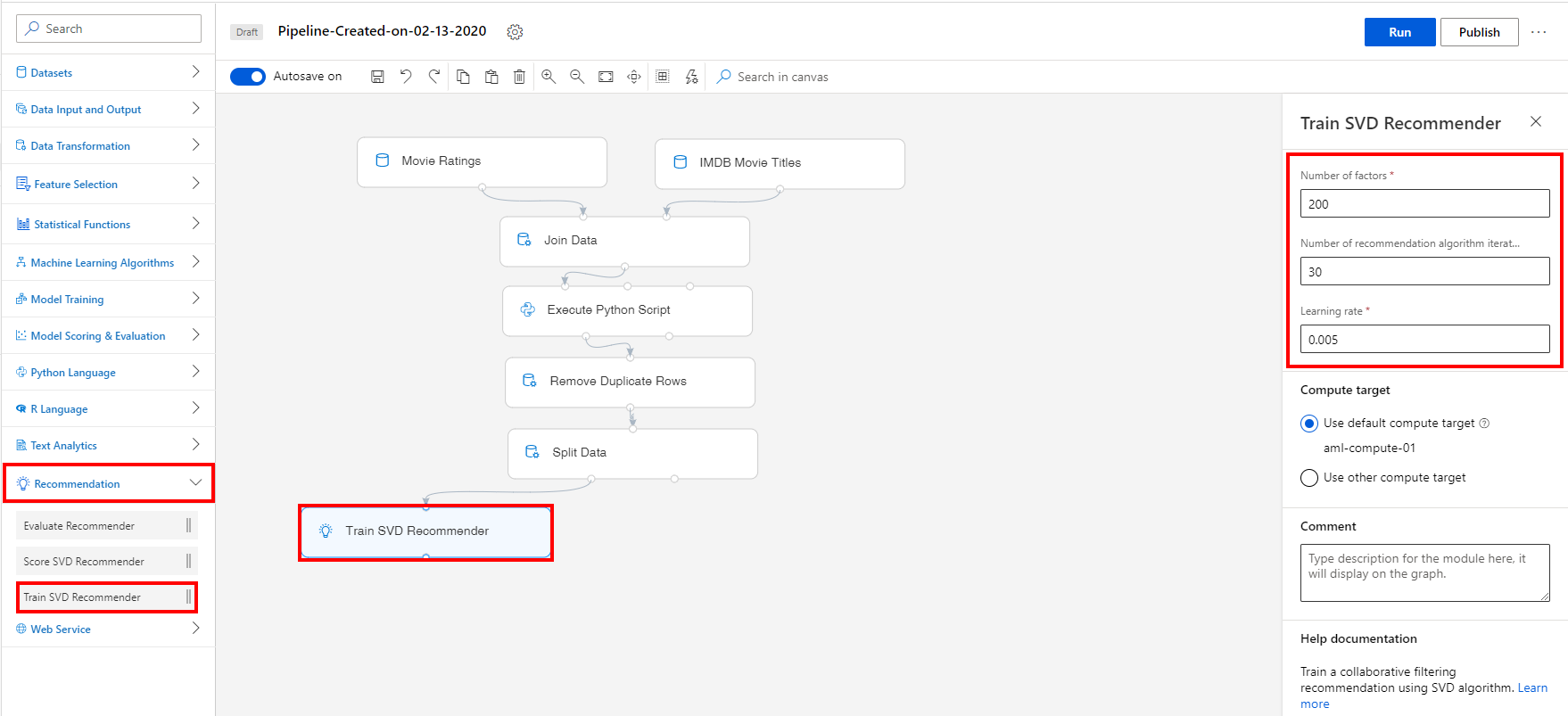
Task 7: Split the dataset into training set (0.5) and test set (0.5)

1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Split Data** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Fraction of rows in the first output dataset: **0.5**
   4. Connect the Dataset to the Split Data module



Task 8: Initialize Recommendation Module

1. Select **Recommendation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Train SVD Recommender** prebuilt module.
   2. Drag and drop the selected module on to the canvas
   3. Connect the first output of the Split Data module to the input of the Train SVD Recommender module
   4. Number of factors: **200**. This option specify the number of factors to use with the recommender. With the number of users and items increasing, it’s better to set a larger number of factors. But if the number is too large, performance might drop.
   5. Number of recommendation algorithm iterations: **30**. This number indicates how many times the algorithm should process the input data. The higher this number is, the more accurate the predictions are. However, a higher number means slower training. The default value is 30.
   6. For Learning rate: **0.001**. The learning rate defines the step size for learning.



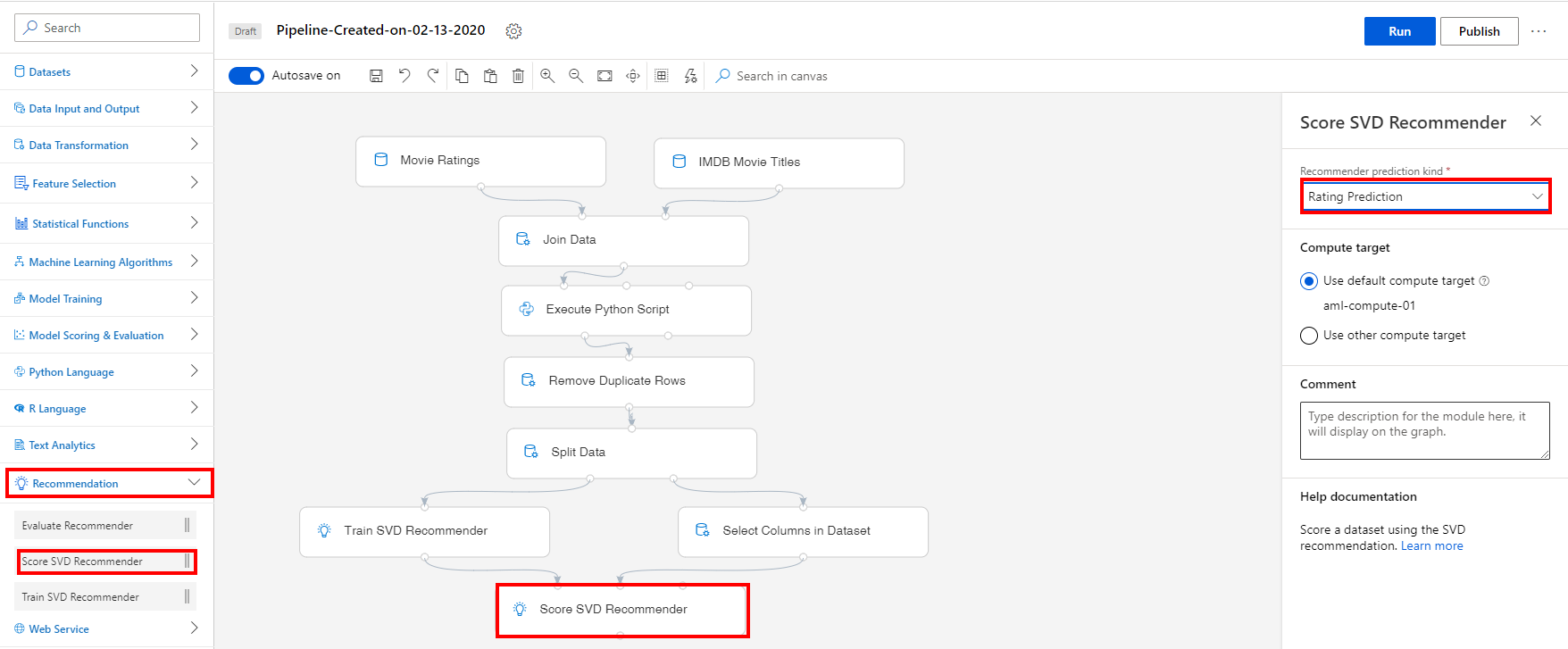
Task 9: Select Columns UserId, Movie Name from the test set

1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Select Columns in Dataset** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the Split Data second output to the input of the Select columns in Dataset module.
   4. Select the **Edit columns** link to open the Select columns editor and then enter the following list of columns to be included in the output dataset: **UserId**, **Movie Name**.

![Image shows the Select Columns in Dataset module added to the canvas` editor.](images/13.png ‘Add Select Columns in Dataset module’)

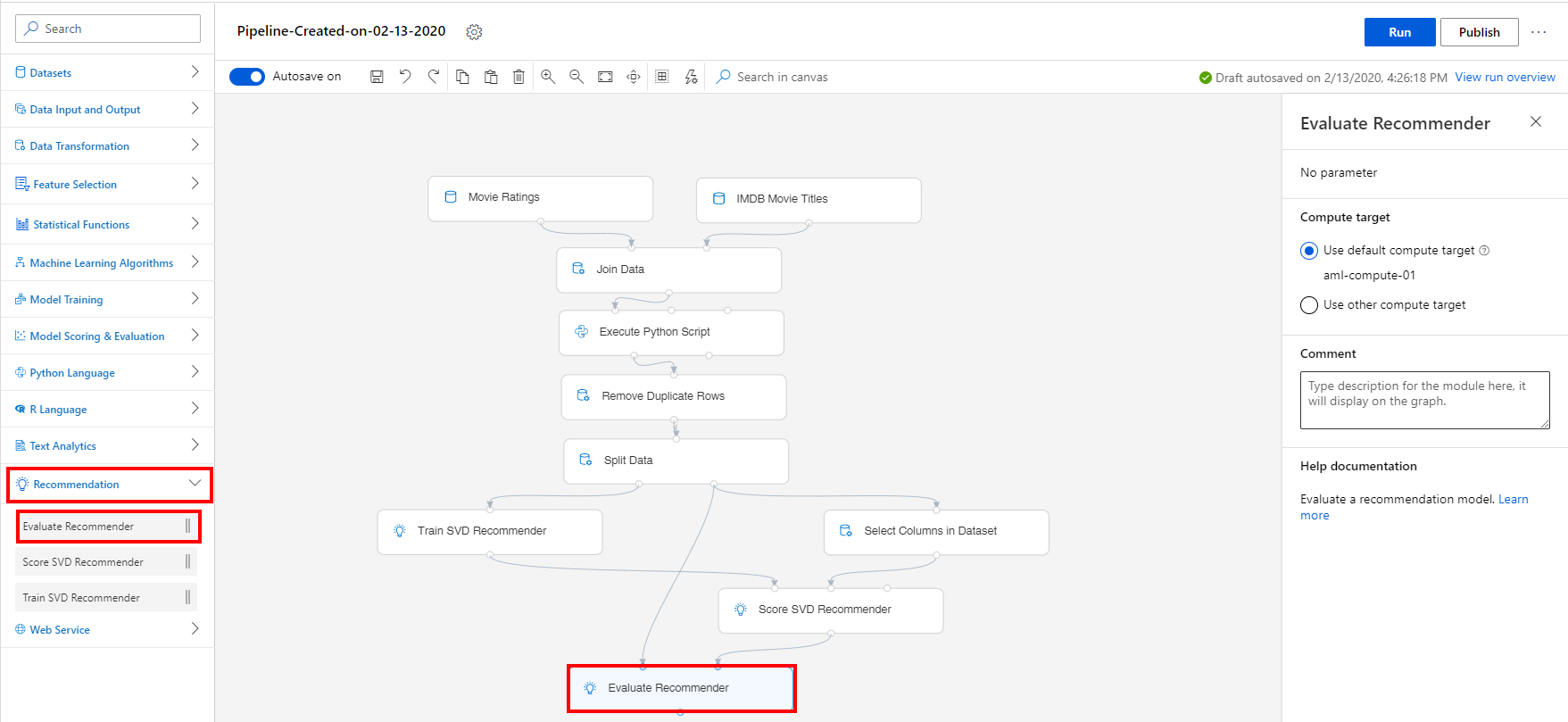
Task 10: Configure the Score SVD Recommender

1. Select **Recommendation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Score SVD Recommender** prebuilt module.
   2. Drag and drop the selected module on to the canvas
   3. Connect the output of the Train SVD Recommender module to the first input of the Score SVD Recommender module, which is the Trained SVD Recommendation input.
   4. Connect the output of the Select Columns in Dataset module to the second input of the Score SVD Recommender module, which is the Dataset to score input.
   5. Select the Score SVD Recommender module on the canvas.
   6. Recommender prediction kind: **Rating Prediction**. For this option, no other parameters are required. When you predict ratings, the model calculates how a user will react to a particular item, given the training data. The input data for scoring must provide both a user and the item to rate.



Task 11: Setup Evaluate Recommender Module

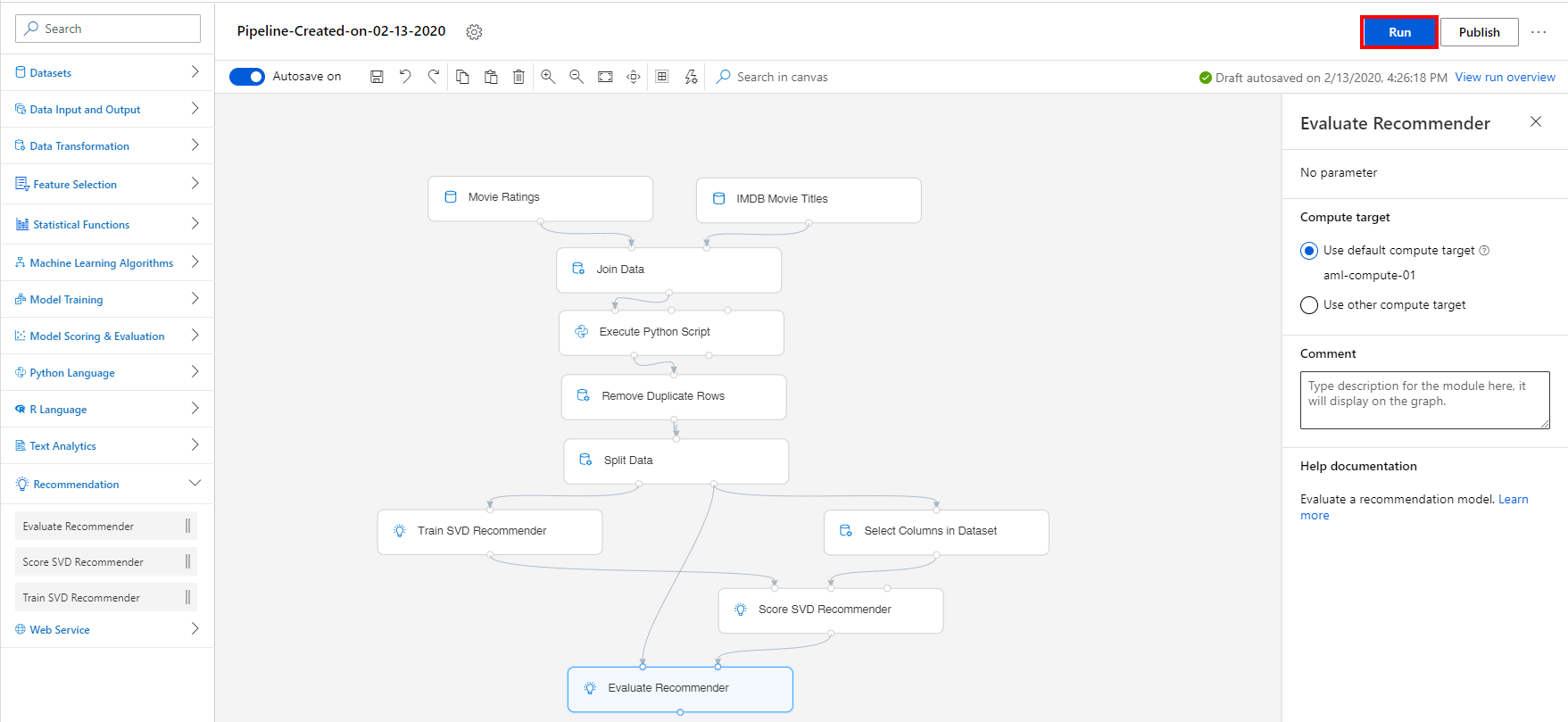
1. Select **Recommendation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Evaluate Recommender** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the Score SVD Recommender module to the second input of the Evaluate Recommender module, which is the Scored dataset input.
   4. Connect the second output of the Split Data module (train set) to the first input of the Evaluate Recommender module, which is the Test dataset input.



Exercise 2: Submit Training Pipeline

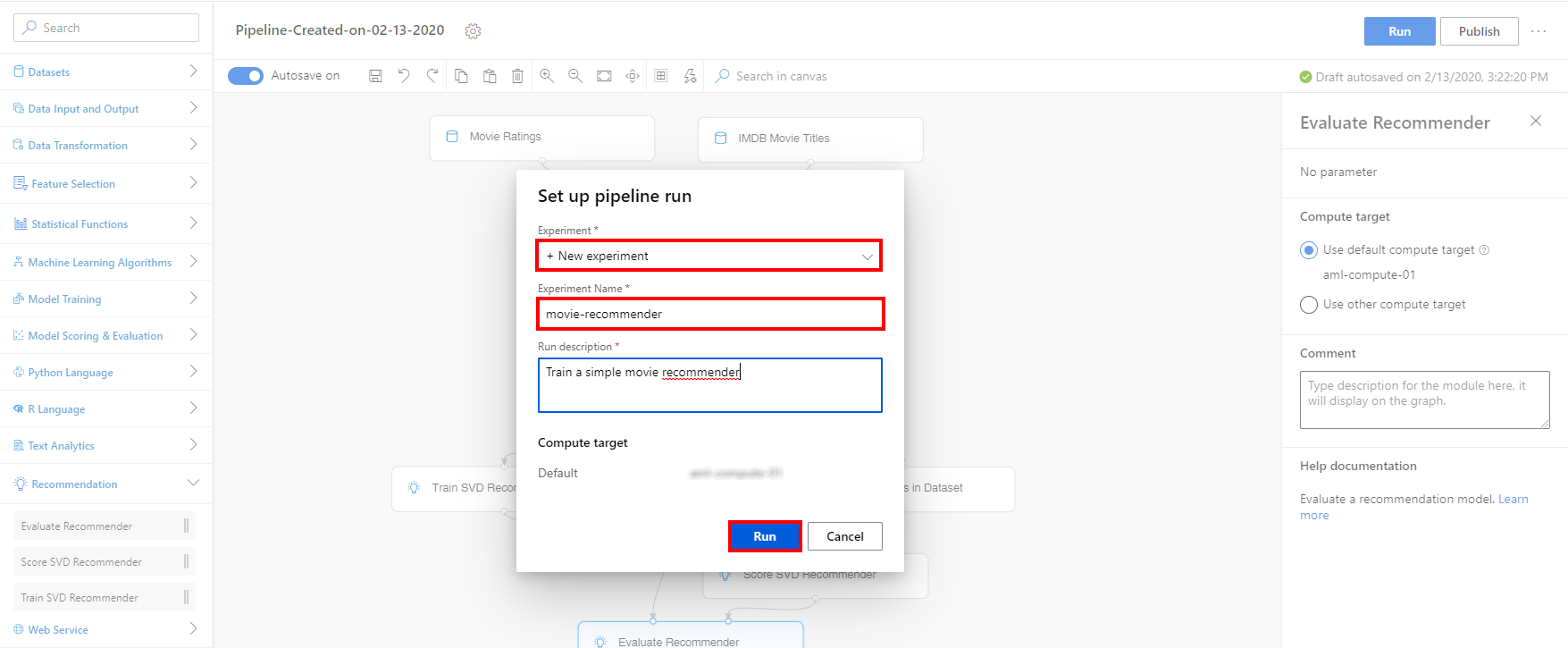
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** on the right corner of the canvas to open the Setup pipeline run editor.



Please note that the button name in the UI is changed from **Run** to **Submit**.

1. In the Setup pipeline run editor, select **Experiment, Create new** and provide New experiment name: **movie-recommender**, and then select **Submit**.

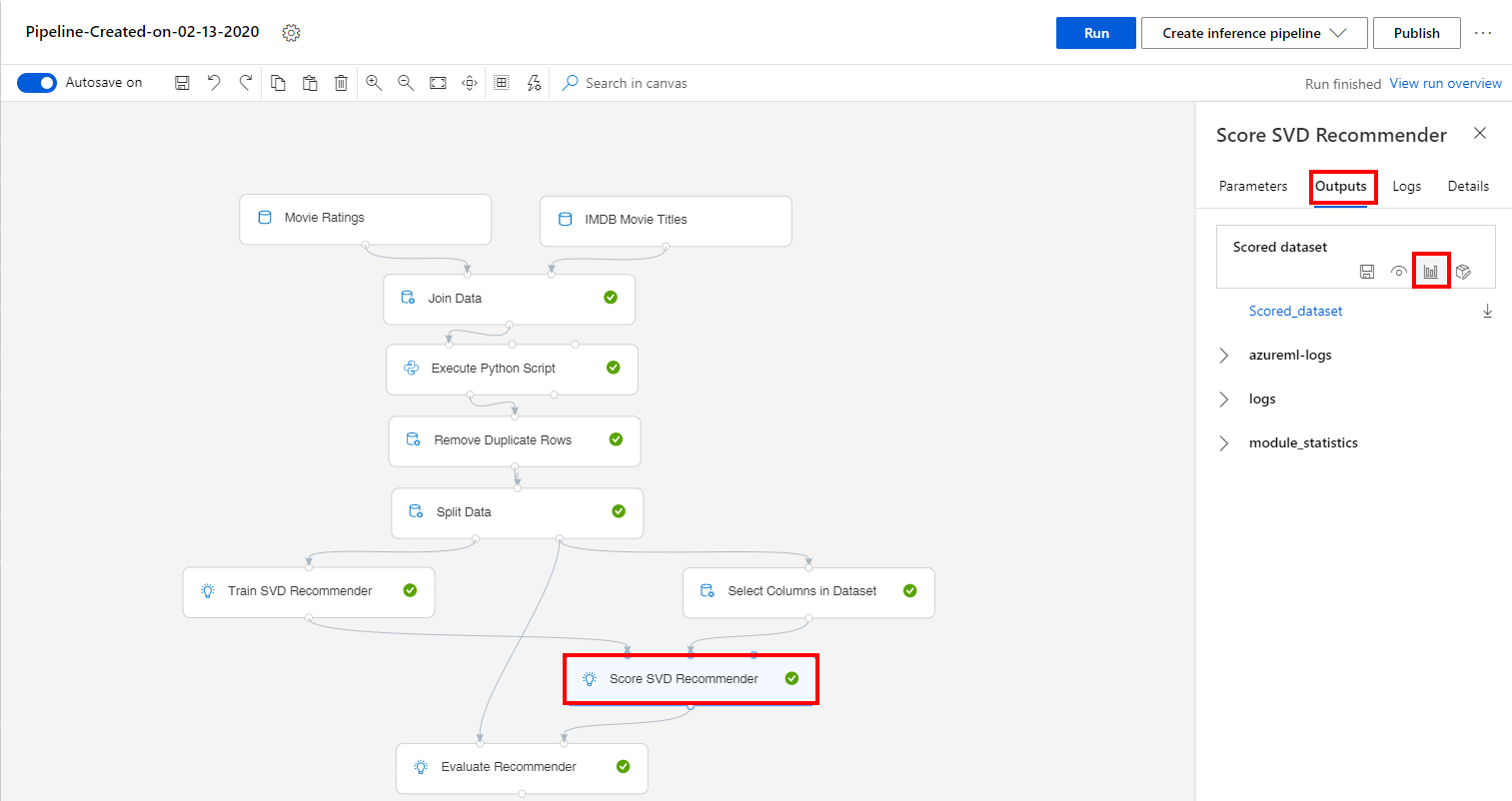


1. Wait for pipeline run to complete. It will take around **20 minutes** to complete the run.
2. While you wait for the model training to complete, you can learn more about the SVD algorithm used in this lab by selecting [Train SVD Recommender](https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/train-svd-recommender).

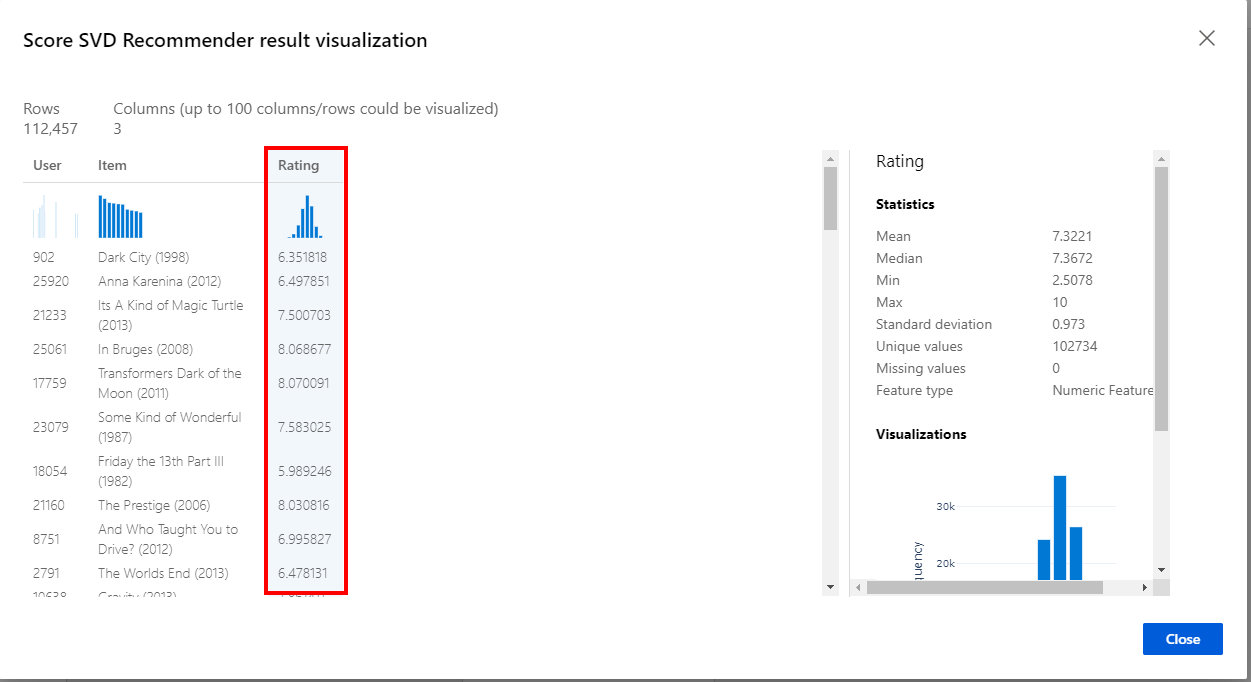
Exercise 3: Visualize Scoring Results

Task 1: Visualize the Scored dataset

1. Select **Score SVD Recommender, Outputs, Visualize** to open the Score SVD Recommender result visualization dialog or just simply right-click the Score SVD Recommender module and select **Visualize Scored dataset**.

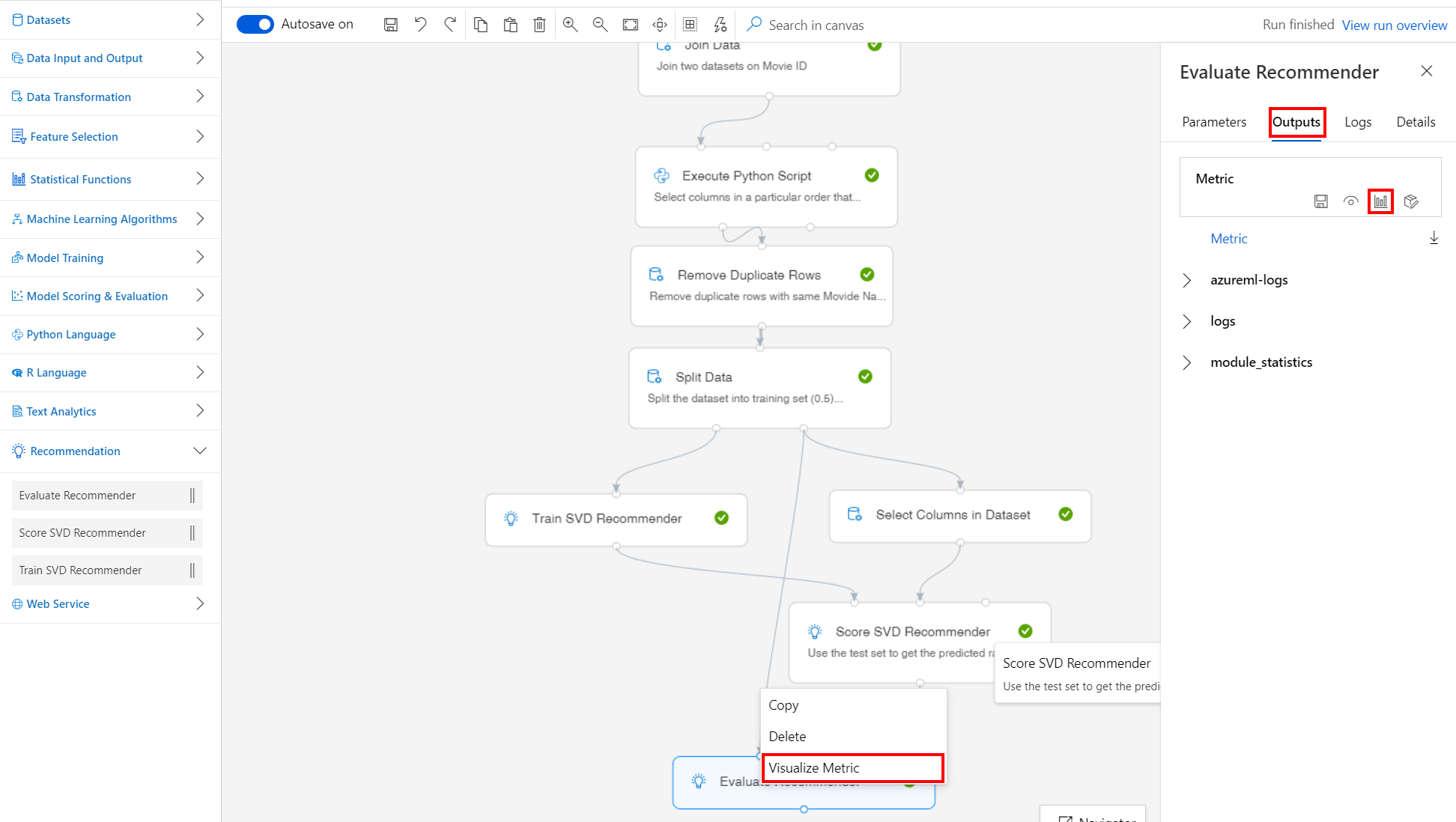


1. Observe the predicted values under the column **Rating**.



Task 2: Visualize the Evaluation Results

1. Select **Evaluate Recommender, Outputs, Visualize** to open the Evaluate Recommender result visualization dialog or just simply right-click the Evaluate Recommender module and select **Visualize Evaluation Results**.



1. Evaluate the model performance by reviewing the various evaluation metrics, such as **Mean Absolute Error**, **Root Mean Squared Error**, etc.



Next Steps

Congratulations! You have trained a simple movie recommender using the prebuilt Recommender modules in the AML visual designer. You can continue to experiment in the environment but are free to close the lab environment tab and return to the Udacity portal to continue with the lesson.