
Microsoft Scholarship Foundation course Nanodegree Program

Labs walkthrough compilation

Compilation of walkthrough made available in the course. All rights reserved to course authors and Udacity.



"Designed by starline / Freepik"



Lab 1 - Train a Linear Regression Model	6
Overview	6
Exercise 1: Register Dataset with Azure Machine Learning studio	7
Exercise 2: Create New Training Pipeline	10
Exercise 3: Submit Training Pipeline	16
Exercise 4: Visualize Training Results	17
Lab 2 - Import, Transform, and Export Data	19
Overview	19
Exercise 1: Import, transform and export data using the Visual Pipeline Authoring Editor	20
Exercise 2: Restructure the data split across multiple files	26
Lab 3 - Create and version a dataset	32
Overview	32
Exercise 1: Register Dataset with Azure Machine Learning studio	33
Exercise 2: Create a version of the existing Dataset	36
Lab 4 - Engineer and select features	42
Overview	42
Exercise 1: Data pre-processing using the Pipeline Authoring Editor	43
Lab 5 - Train and Evaluate a Model	56
Overview	56
Exercise 1: Register Dataset with Azure Machine Learning studio	57
Exercise 2: Create New Training Pipeline	60
Exercise 3: Submit Training Pipeline	67
Exercise 4: Visualize the Evaluation Results	68
Lab 6 - Train a Two-Class Boosted Decision Tree	70
Overview	70
Exercise 1: Register Dataset with Azure Machine Learning studio	71
Exercise 2: Create New Training Pipeline	74
Exercise 3: Submit Training Pipeline	81

Exercise 4: Visualize the Evaluation Results	82
Lab 7 - Train a Simple Classifier with Automated ML	84
Overview	84
Exercise 1: Register Dataset with Azure Machine Learning studio	85
Exercise 2: Setup New Automated Machine Learning Experiment	88
Exercise 3: Start and Monitor Experiment	91
Exercise 4: Review Best Model's Performance	93
Lab 8 - Two-Class Classifiers Performance	96
Overview	96
Exercise 1: Create Training Pipeline	97
Exercise 2: Submit Training Pipeline	101
Exercise 3: Compare Model Performance	102
Lab 9 - Multi-Class Classifiers Performance	105
Overview	105
Exercise 1: Create Training Pipeline	106
Exercise 2: Submit Training Pipeline	109
Exercise 3: Compare Model Performance	110
Lab 10 - Train a Classifier Using Automated Machine Learning	112
Overview	112
Exercise 1: Register Dataset with Azure Machine Learning studio	113
Exercise 2: Setup New Automated Machine Learning Experiment	116
Exercise 3: Start and Monitor Experiment	119
Exercise 4: Review Best Model's Performance	120
Lab 11 - Regressors Performance	123
Overview	123
Exercise 1: Create Training Pipeline	124
Exercise 2: Submit Training Pipeline	127
Exercise 3: Compare Model Performance	128
Lab 12 - Train a Regressor using Automated Machine Learning	130

Overview	130
Exercise 1: Register Dataset with Azure Machine Learning studio	131
Exercise 2: Setup New Automated Machine Learning Experiment	134
Exercise 3: Start and Monitor Experiment	137
Exercise 4: Review Best Model's Performance	138
Lab 13 - Train a Simple Clustering Model	141
Overview	141
Exercise 1: Create New Training Pipeline	142
Exercise 2: Submit Training Pipeline	151
Exercise 3: Visualize the Clustering Results	152
Lab 14 - Train a simple neural net model	154
Overview	154
Exercise 1: Register Dataset with Azure Machine Learning studio	155
Exercise 2: Create New Training Pipeline	158
Exercise 3: Submit Training Pipeline	165
Exercise 4: Visualize Training Results	166
Lab 15 - Train a simple recommender	168
Overview	168
Exercise 1: Create New Training Pipeline	169
Exercise 2: Submit Training Pipeline	179
Exercise 3: Visualize Scoring Results	180
Lab 16 - Train a simple text classifier	183
Overview	183
Exercise 1: Create New Training Pipeline	184
Exercise 2: Submit Training Pipeline	199
Exercise 3: Visualize Training Results	200
Lab 17 - Train a time-series forecasting model using Automated Machine Learning	203
Overview	203

Exercise 1: Creating a model using automated machine learning	204
Lab 18 - Managing a compute instance	213
Overview	213
Exercise 1: Create New Compute Instance	214
Exercise 2: Explore Compute Instances	216
Lab 19 - Train a machine learning model from a managed notebook environment	219
Overview	219
Exercise 1: Run the Notebook for this Lab	220
Lab 20 - Explore experiments and runs	224
Overview	224
Exercise 1: Run the Notebook for this Lab	225
Exercise 2: Open Experiments in the portal	228
Lab 21 - Deploy a trained model as a webservice	233
Overview	233
Exercise 1: Open a sample training pipeline	234
Exercise 2: Real-time inference pipeline	238
Exercise 3: Deploy web service on Azure Kubernetes Service compute	240
Lab 22 - Training and deploying a model from a notebook running in a Compute Instance	244
Overview	244
Exercise 1: Run the Notebook for this Lab	245
Lab 23 - Explaining models: Model interpretability with Azure Machine Learning service	249
Overview	249
Exercise 1: Run the Notebook for this Lab	250

Lab 1 - Train a Linear Regression Model

Overview

Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be using a subset of NYC Taxi & Limousine Commission - green taxi trip records available from [Azure Open Datasets](#). The data is enriched with holiday and weather data. Based on the enriched dataset, we will learn to use the Azure Machine Learning Graphical Interface to process data, build, train, score, and evaluate a regression model to predict NYC taxi fares. To train the model, we will create Azure Machine Learning Compute resource. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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

The screenshot shows the Azure Machine Learning studio landing page. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets (Experiments, Pipelines, Compute, Models, Images, Deployments, Activities), and Getting Started. The main area displays workspace details: Edition: Enterprise, Resource group: [redacted], Location: West Europe, Subscription: [redacted], Subscription ID: [redacted]. It also shows storage, registry, key vault, and application insights links. A central callout box says "Try the new Azure Machine Learning studio" with a "Launch now" button (which is highlighted with a red box) and a "Learn more" link. Below the callout is a "Getting Started" section with a "Get started" button.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124

southcentralus

aml-quickstarts-190124

Get started

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**.

4. From the studio, select **Datasets**, + Create dataset, From web files. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the Azure Machine Learning studio's 'Datasets' view. On the left, there's a sidebar with options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets' (which is selected and highlighted with a red box), 'Experiments', 'Pipelines', 'Models', and 'Endpoints'. The main area is titled 'Datasets' and shows 'Registered datasets' and 'Dataset monitors (Preview)'. Below that is a table with columns 'Version', 'Created on', 'Modified on', 'Properties', and 'Tags'. There are three options in the 'Create dataset' dropdown: 'From local files', 'From datastore', and 'From web files' (which is also highlighted with a red box). At the bottom, it says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampleddata.blob.core.windows.net/data/nyc-taxi/nyc-taxi-sample-data.csv>

6. Provide **nyc-taxi-sample-data** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there's a sidebar with 'Basic info' (selected and highlighted with a red box), 'Settings and preview', 'Schema', and 'Confirm details'. The main panel is titled 'Basic info' and contains the following fields:
 - Web URL *: `https://quickstartsws9073123377.blob.core.windows.net/azure/ml-blobstore-0d1c4218-a5f9-418b-bf55-902b65277b85/quickstarts/nyc-taxi-data/nyc-taxi-sample-data.csv` (this field is highlighted with a red box)
 - Name *: `nyc-taxi-sample-data` (this field is highlighted with a red box)
 - Dataset version: 1
 - Dataset type *: Tabular
 - Description: Dataset description

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Scroll the data preview to right to observe the target column: **totalAmount**. After you are done reviewing the data, select **Next**

Create dataset from web files

Basic info

Settings and preview (selected)

Schema

Confirm details

Settings and preview

These settings were automatically detected. Please verify that the selections were made correctly or update.

File format: Delimited

Delimiter: Comma Example: Field1,Field2,Field3

Encoding: UTF-8

Column headers: All files have same headers (highlighted with a red box)

Skip rows: None

normalizeHolidayName	isPaidTimeOff	snowDep...	precipTime	precipDepth	temperature	totalAmount
None	false	29.06	24.00	3.00	6.19	44.30
None	false	0.00	6.00	0.00	4.57	44.80
None	false	0.00	1.00	0.00	4.38	18.96
None	false	29.06	24.00	3.00	6.19	16.30
None	false	0.00	1.00	0.00	3.85	5.30

Back **Next** **Cancel**

Task 3: Select Columns

- Select columns from the dataset to include as part of your training data. Leave the default selections and select **Next**

quick-starts-ws > Datasets

Datasets

Registered datasets **Dataset monitors**

Create dataset Refresh Unregister

Name	Version

Create dataset from local files

Basic info

Settings and preview (selected)

Schema

Confirm details

Include	Column name	Properties
<input checked="" type="checkbox"/>	Path	Not applicable to select... ▾
<input checked="" type="checkbox"/>	vendorID	Not applicable to select... ▾
<input checked="" type="checkbox"/>	passengerCount	Not applicable to select... ▾
<input checked="" type="checkbox"/>	tripDistance	Not applicable to select... ▾
<input checked="" type="checkbox"/>	hour_of_day	Not applicable to select... ▾
<input checked="" type="checkbox"/>	day_of_week	Not applicable to select... ▾
<input checked="" type="checkbox"/>	day_of_month	Not applicable to select... ▾
<input checked="" type="checkbox"/>	month_num	Not applicable to select... ▾
<input checked="" type="checkbox"/>	normalizeHolidayName	Not applicable to select... ▾

Back **Next** **Cancel**

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' page in the Azure Machine Learning studio. A modal window titled 'Create dataset from local files' is open. The 'Basic info' section contains the file path 'nyc-taxi-sample-data.csv'. The 'File settings' section includes options for 'File format' (Delimited), 'Delimiter' (Comma), 'Encoding' (UTF-8), 'Column headers' (All files have same headers), and 'Skip rows' (None). At the bottom of the modal, there are 'Back', 'Create' (highlighted with a red box), and 'Cancel' buttons.

Exercise 2: Create New Training Pipeline

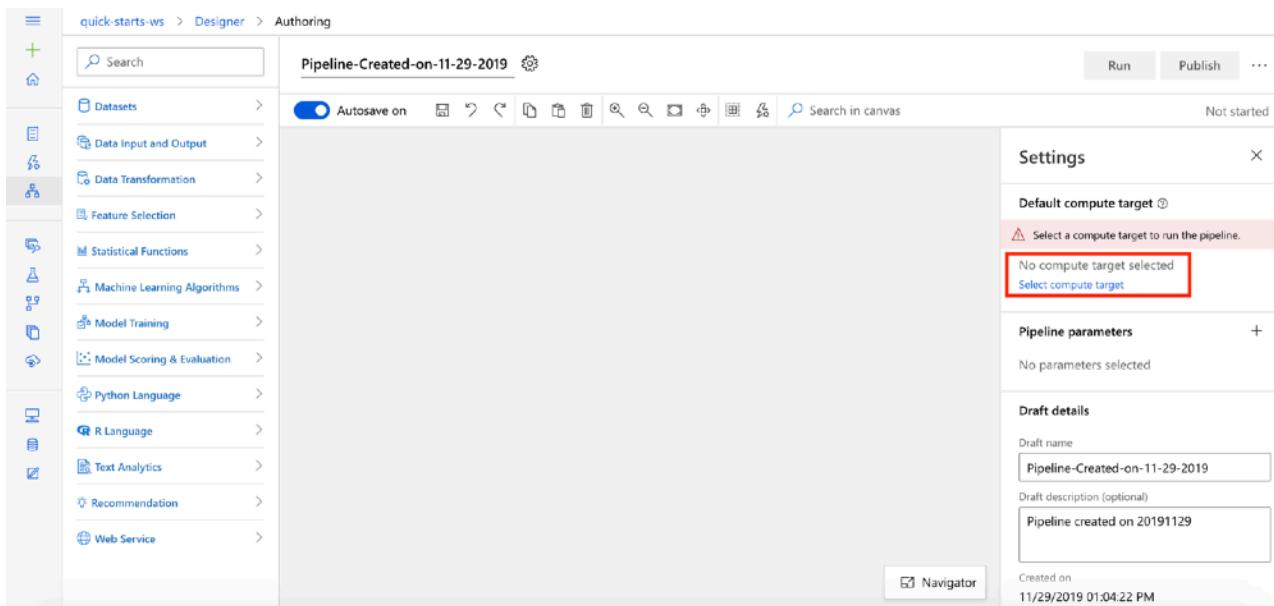
Task 1: Open Pipeline Authoring Editor

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

The screenshot shows the 'Designer' page in the Azure Machine Learning studio. The 'New pipeline' section has a large red box around the '+' icon. Below it are five sample pipeline cards: 'Easy-to-use prebuilt modules', 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', and 'Sample 4: Binary Classification with custom Python script - ...'. The 'Pipelines' section shows a table with columns 'Name', 'Pipeline type', 'Updated on', and 'Created by'. A yellow folder icon is visible at the bottom center.

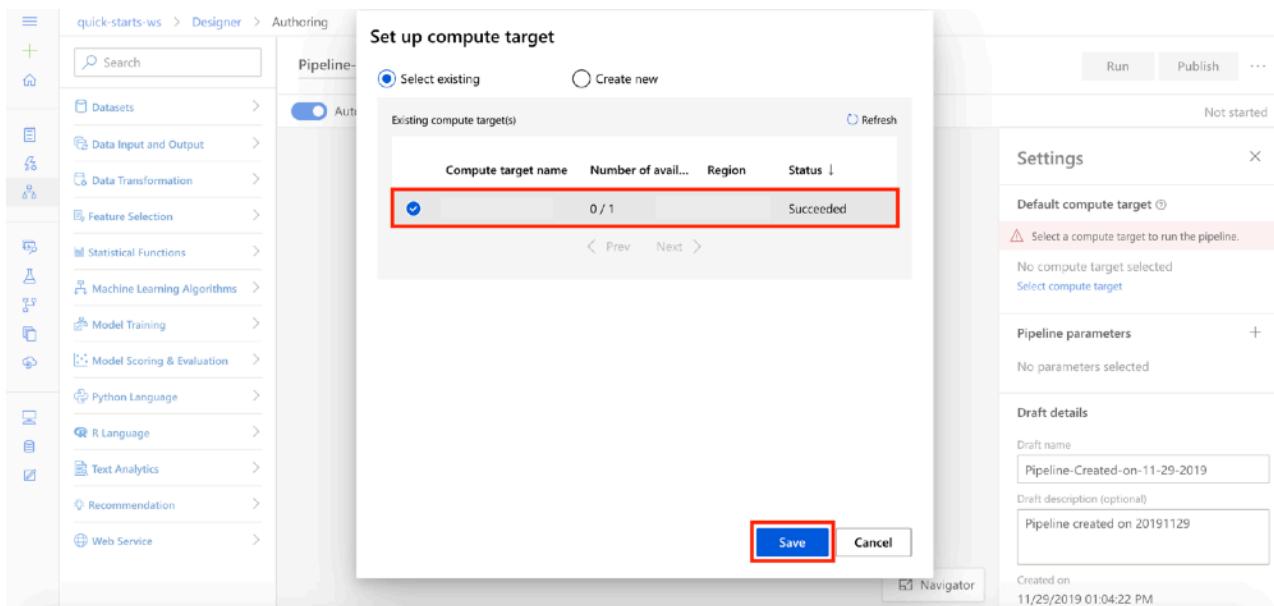
Task 2: Setup Compute Target

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



2. 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 Dataset

1. Select **Datasets** section in the left navigation. Next, select **My Datasets**, **nyc-taxi-sample-data** and drag and drop the selected dataset on to the canvas.

The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, the navigation pane has 'Datasets' selected. A dataset named 'nyc-taxi-sample-data-vxiu2s...' is highlighted with a red box and labeled '1'. In the center canvas, another dataset is also highlighted with a red box and labeled '2'. A tooltip 'Dataset output: DataFrameDirectory' is visible below it. On the right, the properties panel for the dataset shows details like ID, Dataset name, and Datastore name. The 'Outputs' tab is selected.

Task 4: Split Dataset

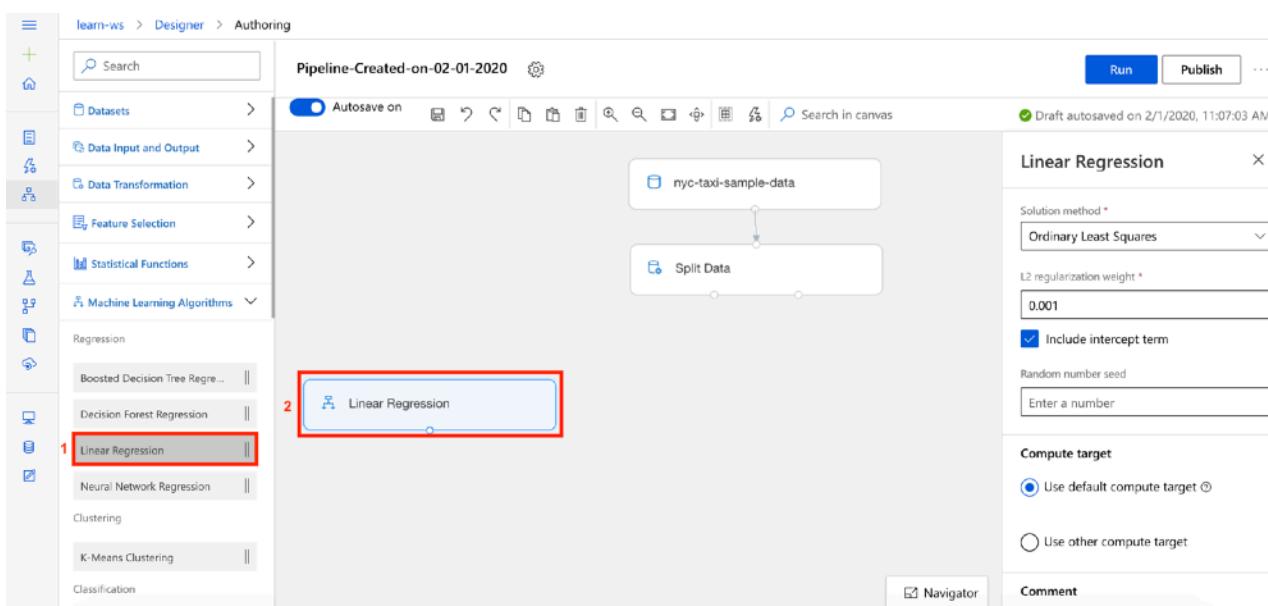
1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Split Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Fraction of rows in the first output dataset: **0.7**
 4. Connect the **Dataset** to the **Split Data** module

The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, the 'Data Transformation' section is selected, and the 'Split Data' module is highlighted with a red box and labeled '1'. In the center canvas, a 'Split Data' module is placed and highlighted with a red box and labeled '2'. A connection line from the dataset on the canvas is connected to the module. The module's properties panel on the right is configured with 'Split Rows' as the splitting mode, '0.7' as the fraction of rows in the first output dataset, and 'Randomized split' checked. Other settings like Random seed and Compute target are also shown.

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.

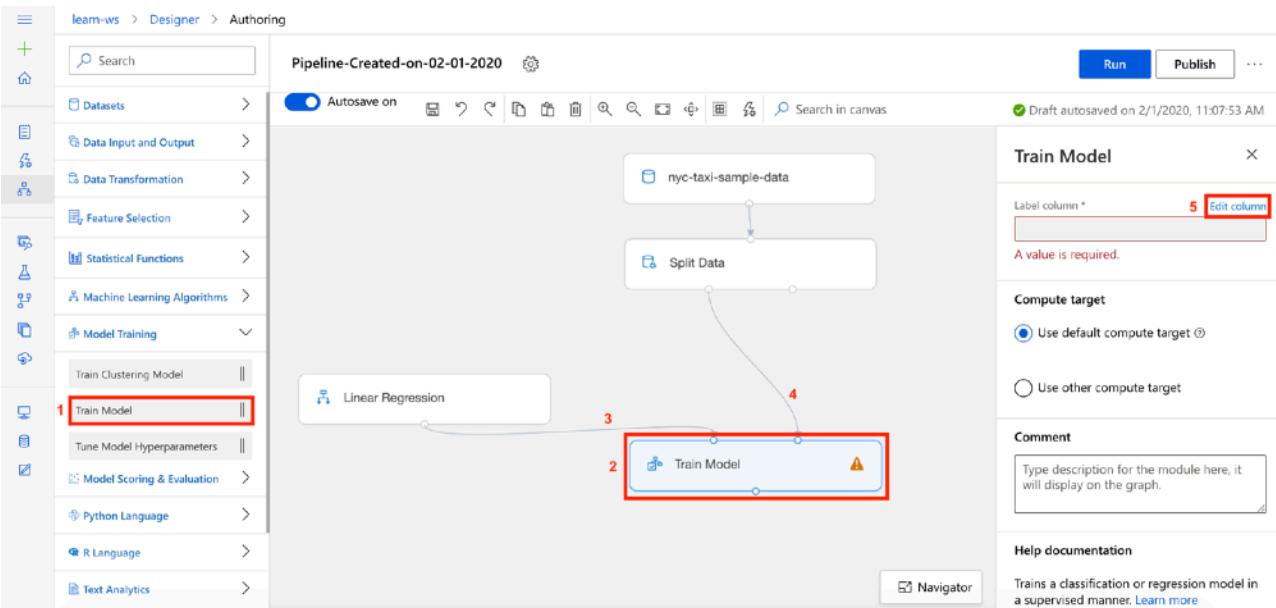
Task 5: Initialize Regression Model

1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Linear Regression** prebuilt module
 2. Drag and drop the selected module on to the canvas

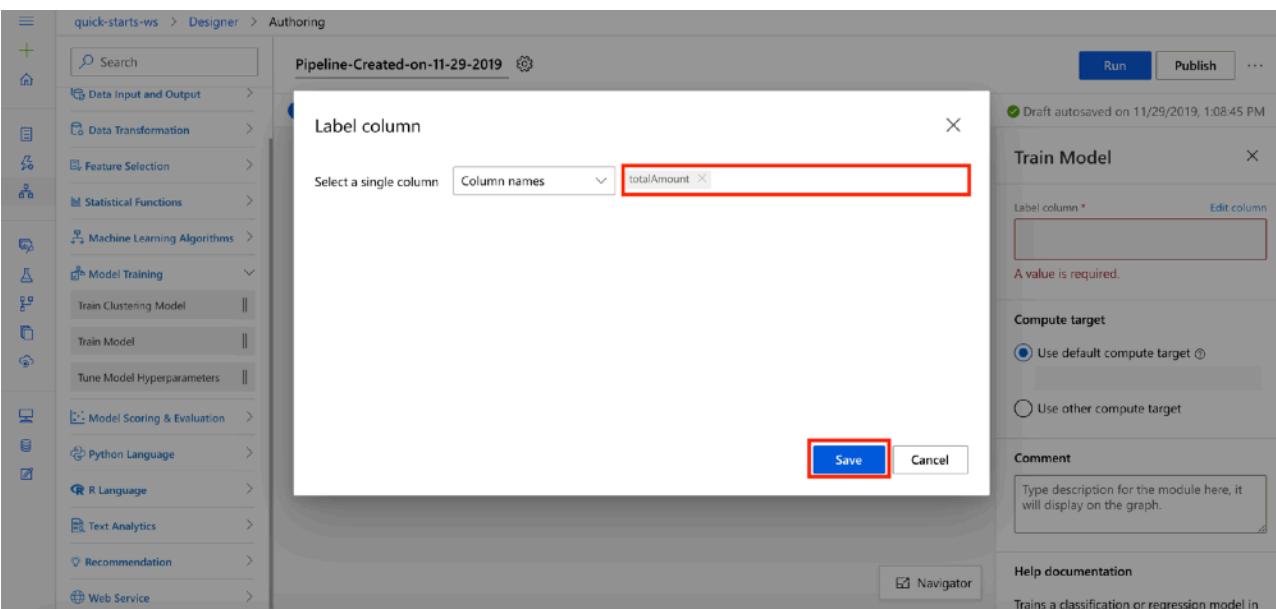


Task 6: Setup Train Model Module

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Linear Regression** module to the first input of the **Train Model** module
 4. Connect the first output of the **Split Data** module to the second input of the **Train Model** module
 5. Select the **Edit column** link to open the **Label column** editor

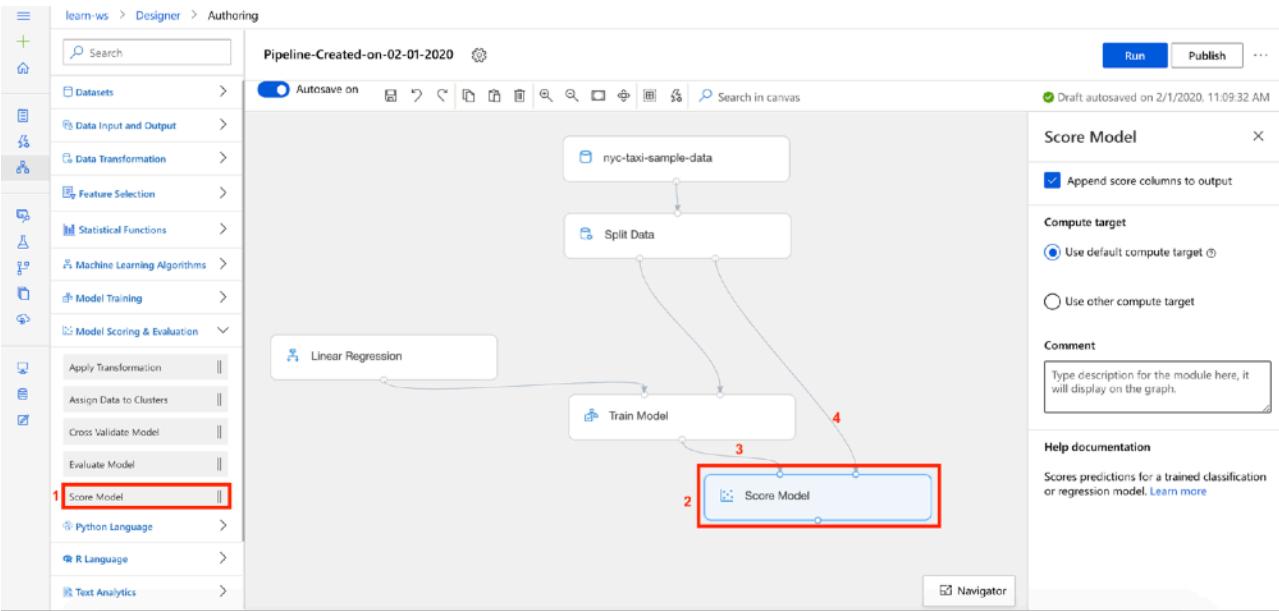


2. The **Label column** editor allows you to specify your **Label or Target column**. Type in the label column name **totalAmount** and then select **Save**.



Task 7: Setup Score Model Module

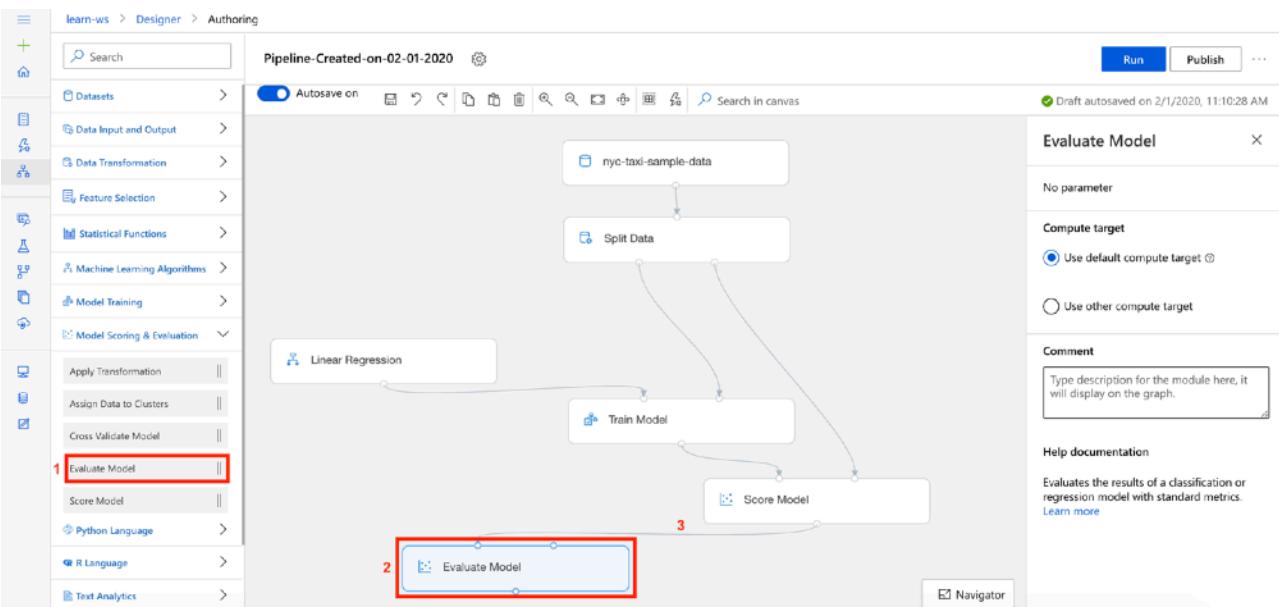
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Train Model** module to the first input of the **Score Model** module
 4. Connect the second output of the **Split Data** module to the second input of the **Score Model** module



Note that **Split Data** module will feed data for both model training and model scoring. The first output (0.7 fraction) will connect with the **Train Model** module and the second output (0.3 fraction) will connect with the **Score Model** module.

Task 8: Setup Evaluate Model Module

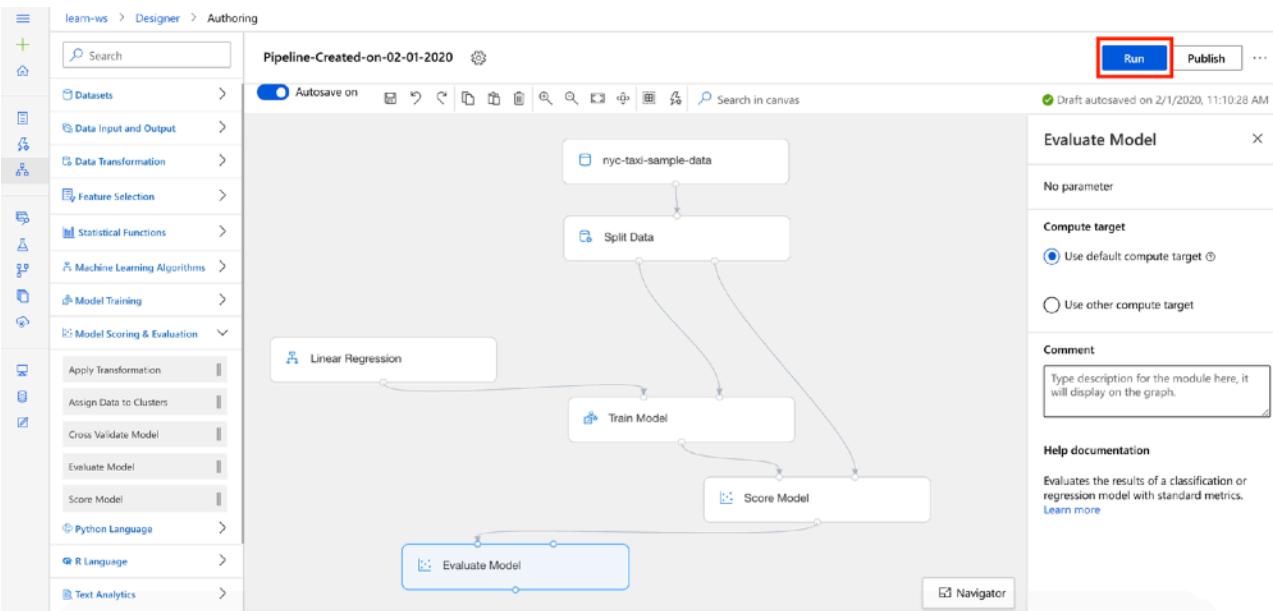
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Score Model** module to the first input of the **Evaluate Model** module



Exercise 3: Submit Training Pipeline

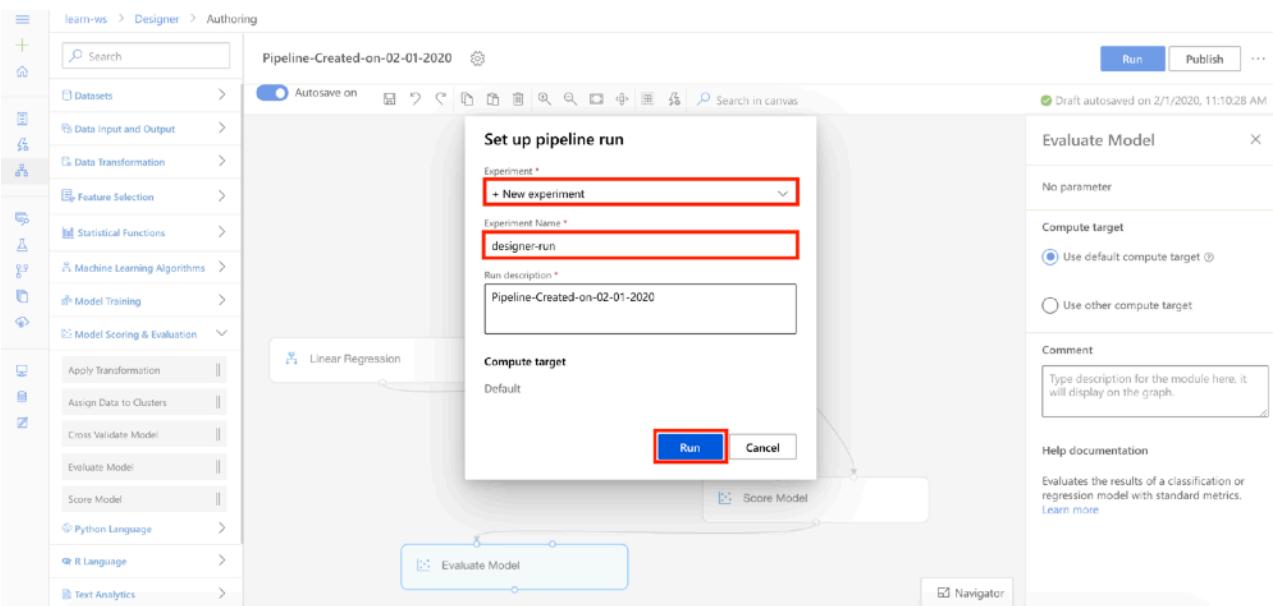
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



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

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

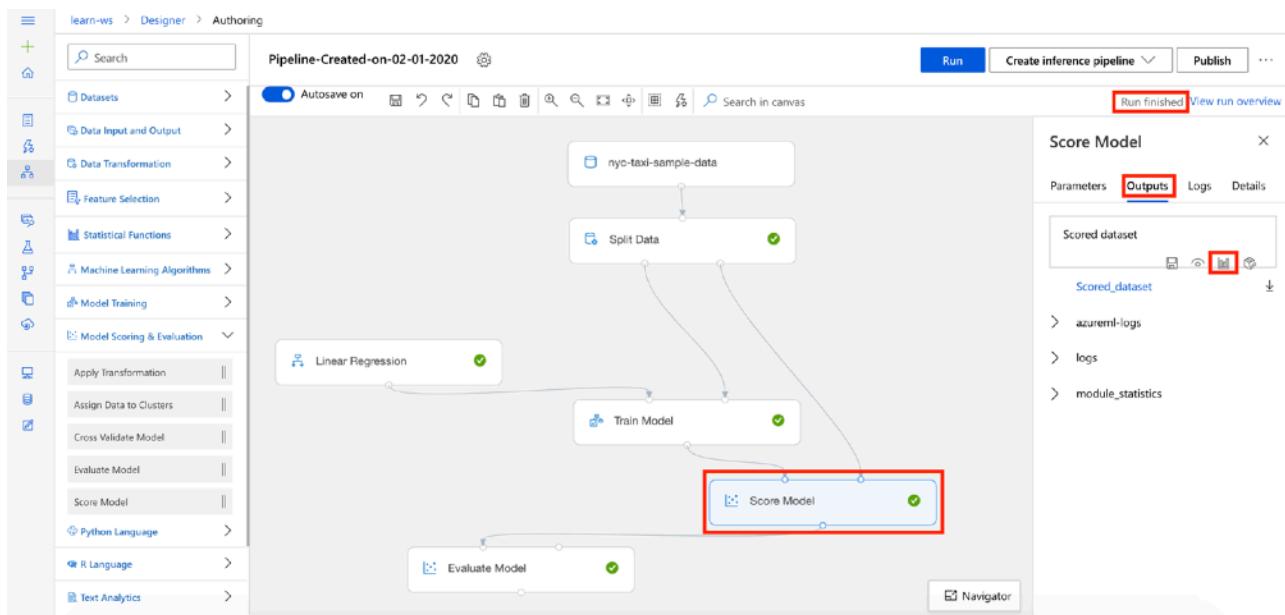


3. Wait for pipeline run to complete. It will take around **8 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the training algorithm used in this lab by selecting **Linear Regression module**.

Exercise 4: Visualize Training Results

Task 1: Visualize the Model Predictions

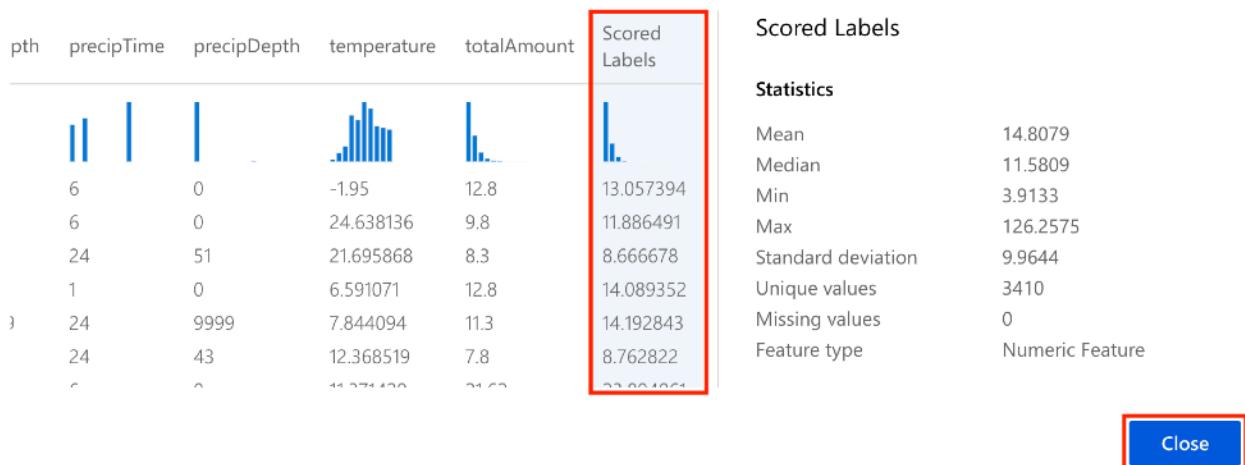
1. Select **Score Model, Outputs, Visualize** to open the **Score Model result visualization** dialog.



2. Observe the predicted values under the column **Scored Labels**. You can compare the predicted values (**Scored Labels**) with actual values (**totalAmount**).

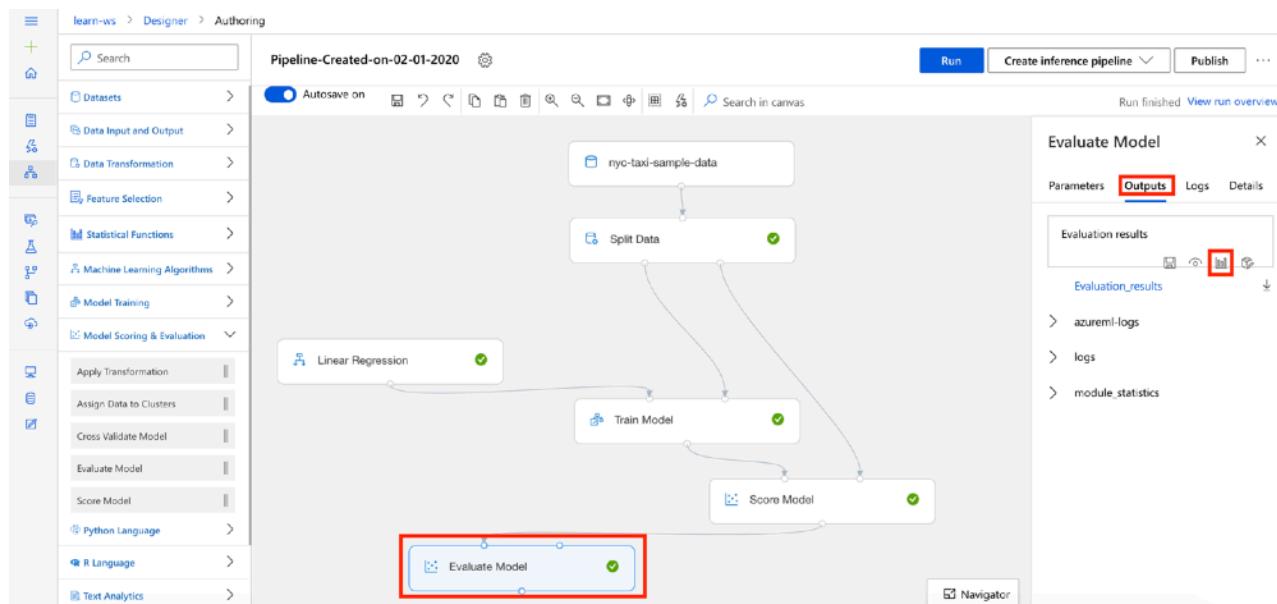
Score Model result visualization

Rows Columns (up to 100 columns/rows could be visualized)
3,520 15



Task 2: Visualize the Evaluation Results

- Select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog.



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

The screenshot shows the 'Evaluate Model result visualization' dialog. At the top, it says 'Evaluate Model result visualization' and has a close button. Below that is a table with 'Rows' (1) and 'Columns' (5). The columns are labeled 'Mean_Absolute_Error', 'Root_Mean_Squared_Error', and 'Relative_Squared_Error'. The values are 2.151788, 3.918504, and 0.144897 respectively. A red box highlights the first two columns. To the right, there's a 'Root_Mean_Squared_Error' section with a 'Statistics' table. The statistics are:

Statistics	Root_Mean_Squared_Error
Mean	3.9185
Median	3.9185
Min	3.9185
Max	3.9185
Standard deviation	-
Unique values	1
Missing values	0
Feature type	Numeric Feature

At the bottom right is a 'Close' button with a red box around it.

Next Steps

Congratulations! You have trained and evaluated your first machine learning model. 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.

Lab 2 - Import, Transform, and Export Data

Overview

In this lab you learn how to import your own data in the designer to create custom solutions. There are two ways you can import data into the designer in Azure Machine Learning Studio:

- Azure Machine Learning datasets
Register datasets in Azure Machine Learning to enable advanced features that help you manage your data.
- Import Data module
Use the Import Data module to directly access data from online datasources.

The first approach will be covered later in the [next lab](#), which focuses on registering and versioning a dataset in Azure Machine Learning studio.

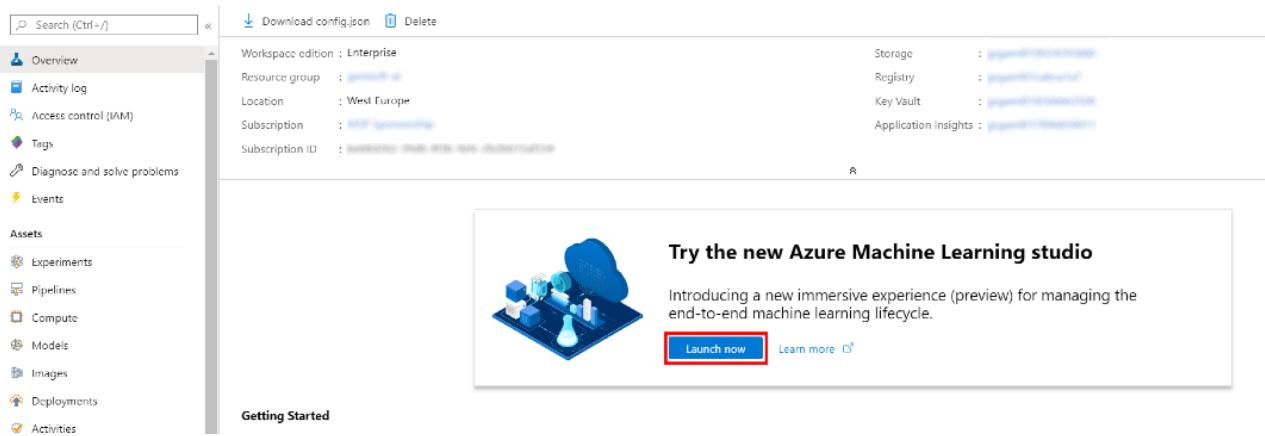
While the use of datasets is recommended to import data, you can also use the Import Data module from the designer. Data comes into the designer from either a **Datastore** or from **Tabular Datasets**. Datastores will be covered later in this course, but just for a quick definition, you can use Datastores to access your storage without having to hard code connection information in your scripts. As for the second option, the Tabular datasets, the following datasources are supported in the designer: Delimited files, JSON files, Parquet files or SQL queries.

The following exercise focuses on the Import Data module to load data into a machine learning pipeline from several datasets that will be merged and restructured. We will be using some sample data from the UCI dataset repository to demonstrate how you can perform basic data import transformation steps with the modules available in Azure Machine Learning designer.

Exercise 1: Import, transform and export data using the Visual Pipeline Authoring Editor

Task 1: Open Pipeline Authoring Editor

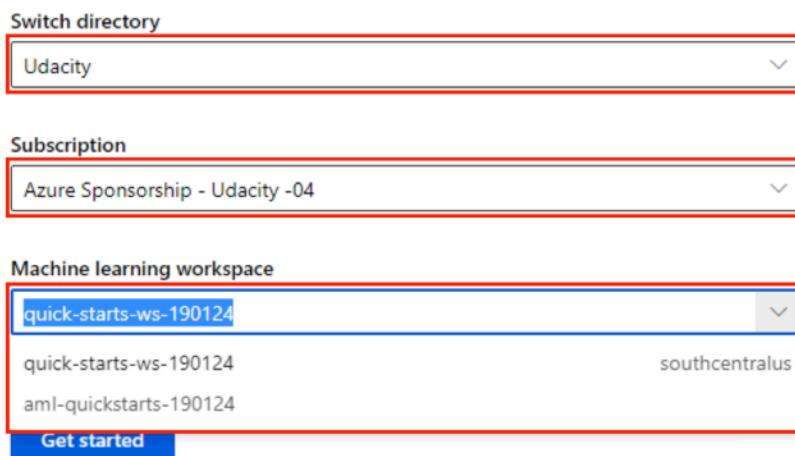
1. In [Azure portal](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



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**.

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

The screenshot shows the Azure Machine Learning Designer interface. On the left, there's a navigation sidebar with options like 'New', 'Home', 'Designer' (which is highlighted with a red box), 'Notebooks', 'Automated ML', 'Datasets', 'Experiments', 'Pipelines', 'Models', 'Endpoints', 'Compute', 'Datastores', and 'Compute targets'. The main area is titled 'Designer' and 'New pipeline'. It features a large red-bordered box around the '+ Add module' button. Below it are four sample pipelines: 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', and 'Sample 4 with cust...'. A search bar at the bottom right says 'Search to filter items...'.

Task 2: Setup Compute Target

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

The screenshot shows the Pipeline-Authoring interface. On the left, there's a sidebar with 'Search', 'Datasets', 'Data Input and Output' (which is expanded, showing 'Enter Data Manually', 'Export Data', 'Import Data', 'Data Transformation', 'Feature Selection', and 'Statistical Functions'), and a 'Run' button. The main area shows a pipeline named 'Pipeline-Created-on-02-10-2020'. On the right, there's a 'Settings' tab with a red box around the 'Default compute target' section, which contains the message 'Select a compute target to run the pipeline.' and a link 'Select compute target'. Other sections include 'Pipeline parameters' (No parameters selected).

2. In the **Set up compute target** editor, select the existing compute target, 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.

Set up compute target

Select existing Create new

Existing compute target(s) Refresh

Compute target name	Number of avail...	Region	Status ↓
my-compute	1 / 1	southcent...	Succeeded

< Prev Next >

Save Cancel

Task 3: Import data from Web URL

1. Select **Data Input and Output** section in the left navigation. Next, select **Import Data** and drag and drop the selected module on to the canvas.

The screenshot shows the Azure Machine Learning Studio interface. The left sidebar is open, displaying various sections like New, Home, Author, Notebooks, Automated ML, Designer, Datasets, Experiments, Pipelines, Models, Endpoints, Compute, Datastores, and Data Labeling. The 'Designer' section is currently selected. Within the Designer section, the 'Data Input and Output' subsection is highlighted with a red box. Inside this subsection, the 'Import Data' module is also highlighted with a red box. The main workspace is titled 'Pipeline-Created-on-02-10-2020' and contains a toolbar with various icons for file operations and search.

2. In the **Import data** panel on the right, select the **URL via HTTP** option in the **Data Source** drop-down and provide the following **Data source URL** for the first CSV file you will import in your pipeline: <https://introtomlsampleddata.blob.core.windows.net/data/crime-data/crime-dirty.csv>

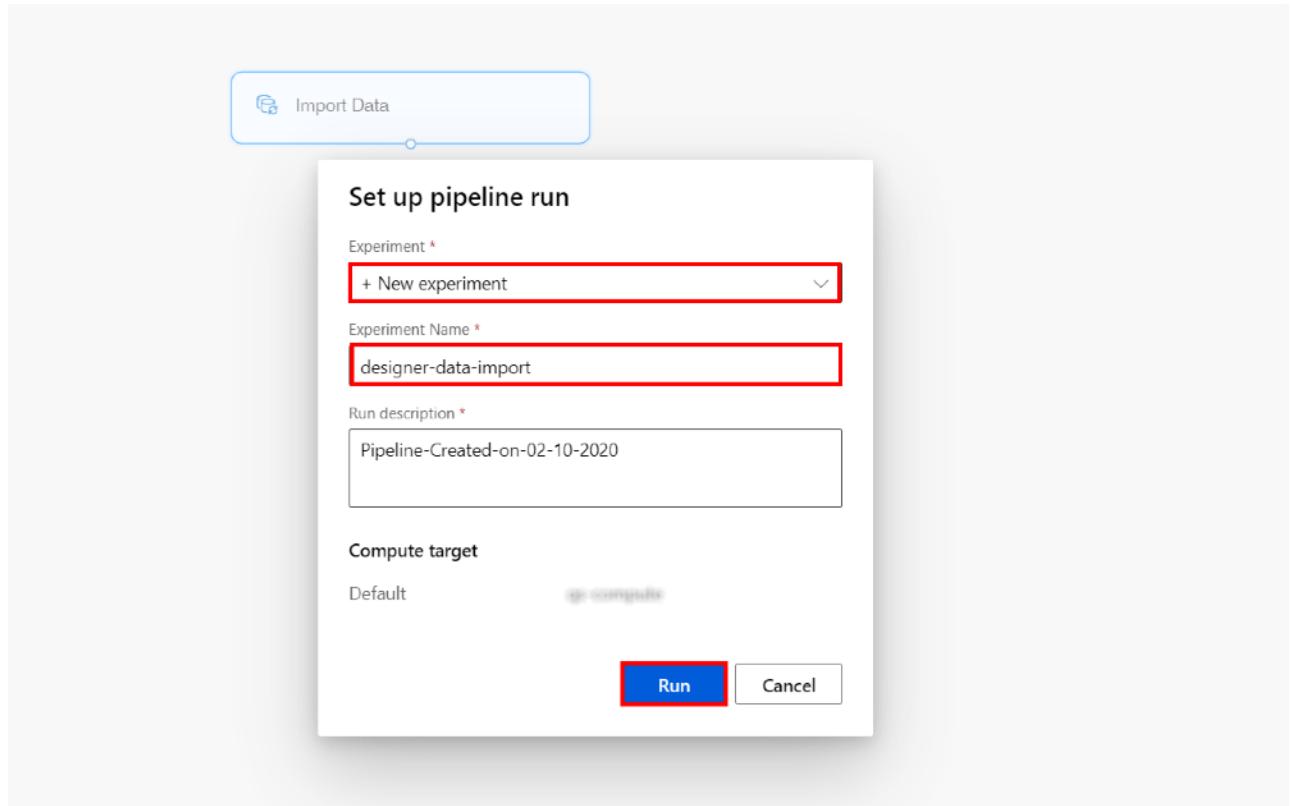
The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, there's a sidebar with various modules like Datasets, Data Input and Output, Data Transformation, Feature Selection, etc. In the center, there's a pipeline graph with a single module labeled "Import Data". To the right of the graph is the "Import Data" configuration pane. The "Data source" dropdown is set to "URL via HTTP". The "Data source URL" input field contains the value "https://introtomlsampleddata.blob.core.windows.net/data/crime-data/crime-dirty.csv". A red box highlights the "Preview schema" button. Below it, the "Compute target" section has a radio button selected for "Use default compute target". Another red box highlights this selection. At the bottom of the pane, there's a "Comment" section and a "Help documentation" link.

3. Select the **Preview schema** to filter the columns you want to include. You can also define advanced settings like Delimiter in **Parsing options**. Select **Save** to close the dialog.

This screenshot shows the "Schema preview" dialog box open in front of the "Import Data" configuration pane. The dialog allows selecting columns from a list. The "Path" column has an empty checkbox. The "ID", "Case Number", "Date", "Block", "IUCR", and "Primary Type" columns have checked checkboxes. The "Type" column shows dropdown menus for each selected column. At the bottom of the dialog are "Save" and "Cancel" buttons. The background "Import Data" pane remains visible, showing the same configuration as the previous screenshot.

Task 4: Create Experiment and Submit Pipeline

1. Back to the pipeline canvas, select **Submit** on the top right corner to open the **Setup pipeline run editor**.
2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: designer-data-import**, and then select **Submit**.

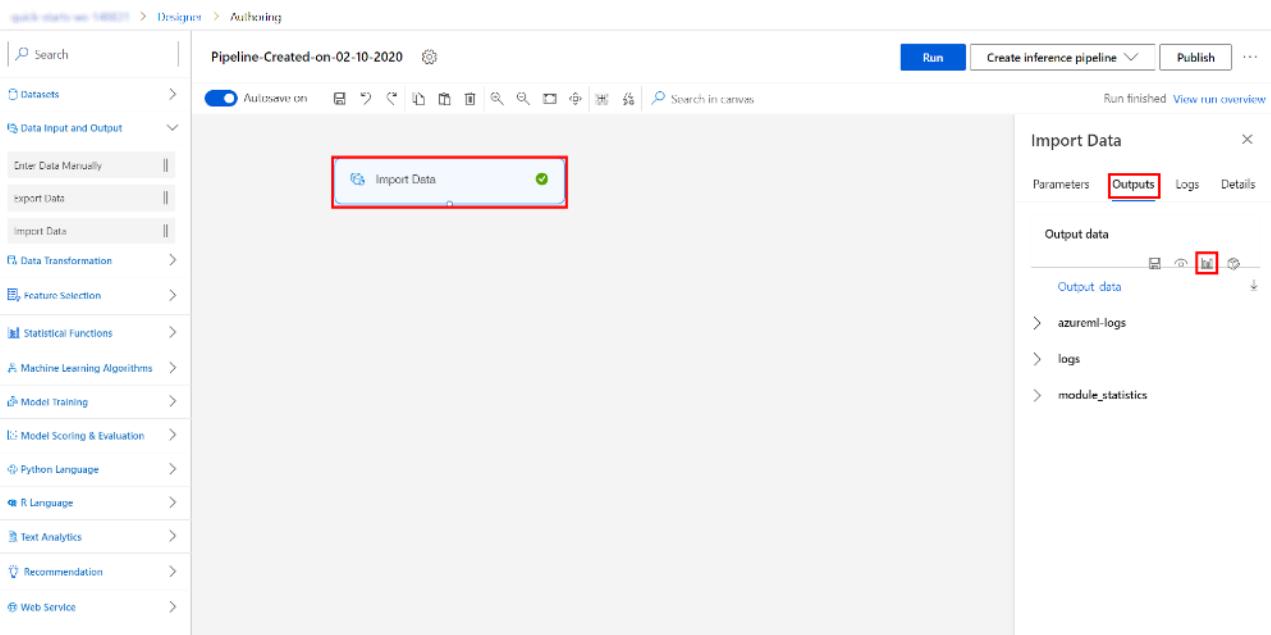


Please note that the button name in the UI is changed from **Run** to **Submit**.

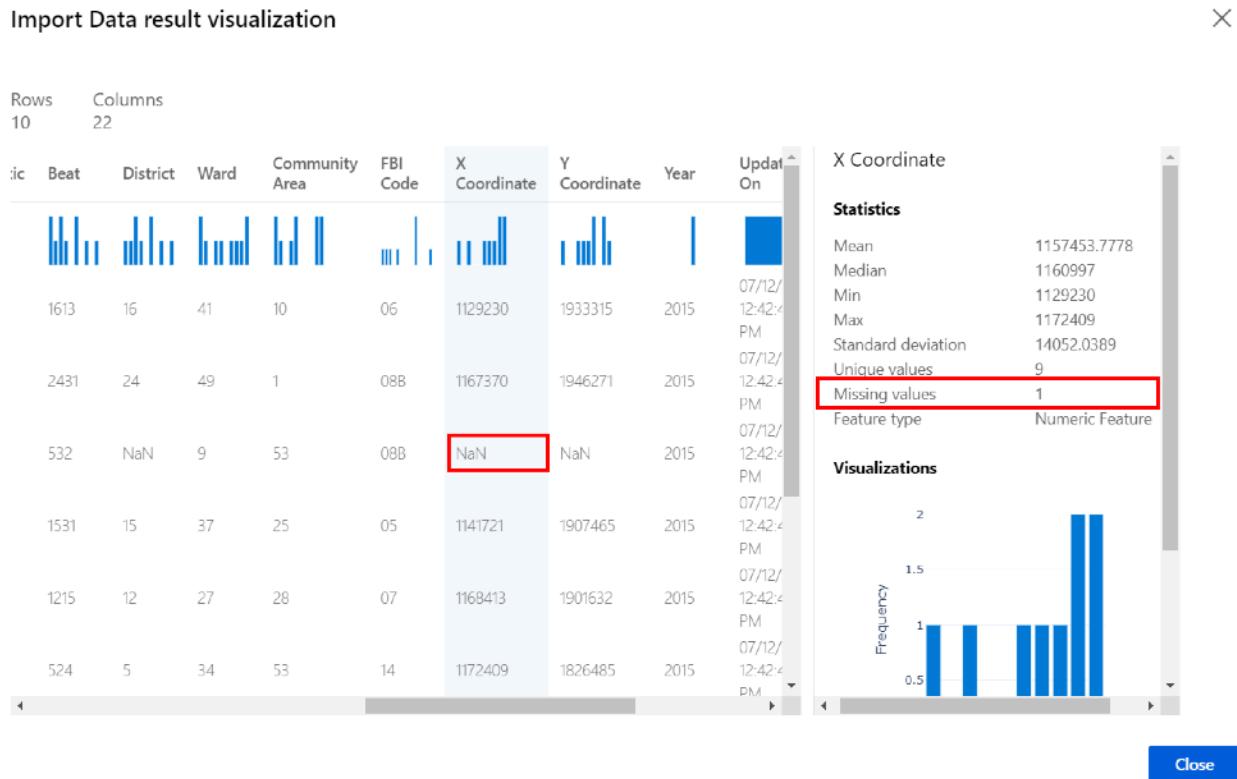
3. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.

Task 5: Visualize Import Data results

1. Select the **Import Data** module on the canvas and then select **Outputs** on the right pane. Click on the **Visualize** icon to open the **Import Data result visualization** dialog.



2. In the **Import Data result visualization** dialog take some moments to explore all the metadata that is now available to you, such as: number of rows, columns, preview of data and for each column you select you can observe: **Mean**, **Median**, **Min**, **Max** and also number of **Unique Values** and **Missing Values**. Data profiles help you glimpse into the column types and summary statistics of a dataset. Scroll right and select the **X Coordinate** column. Notice the **Nan** value on the third row in the preview table and check the **Missing values** number in the **Statistics** section.



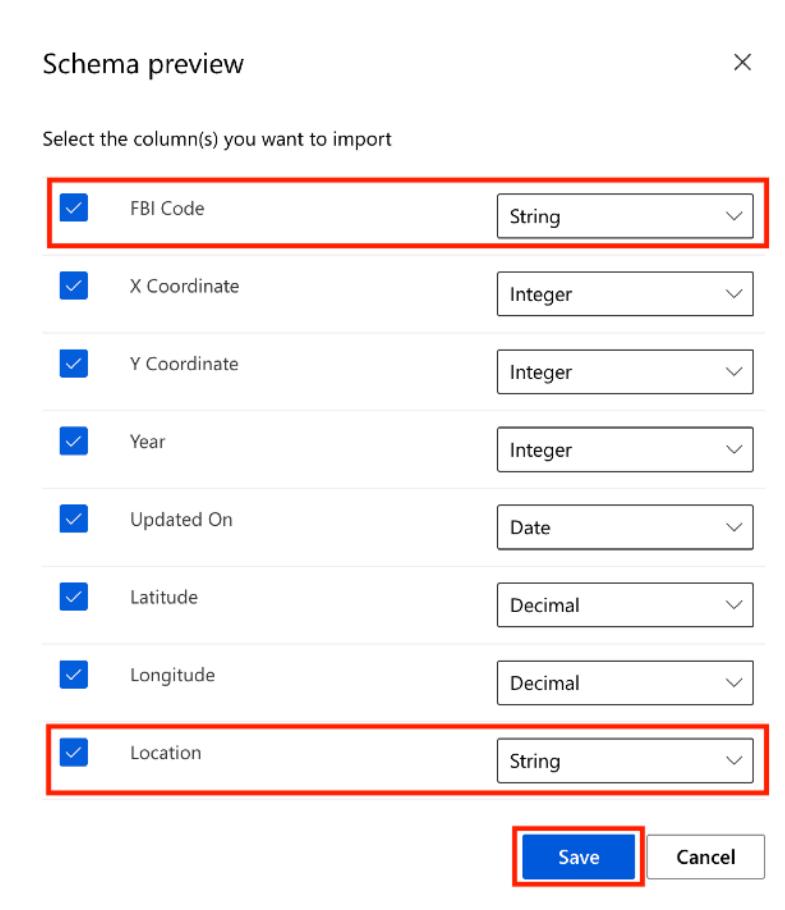
- Select **Close** to return to the pipeline designer canvas where you can continue the data import phase.

Exercise 2: Restructure the data split across multiple files

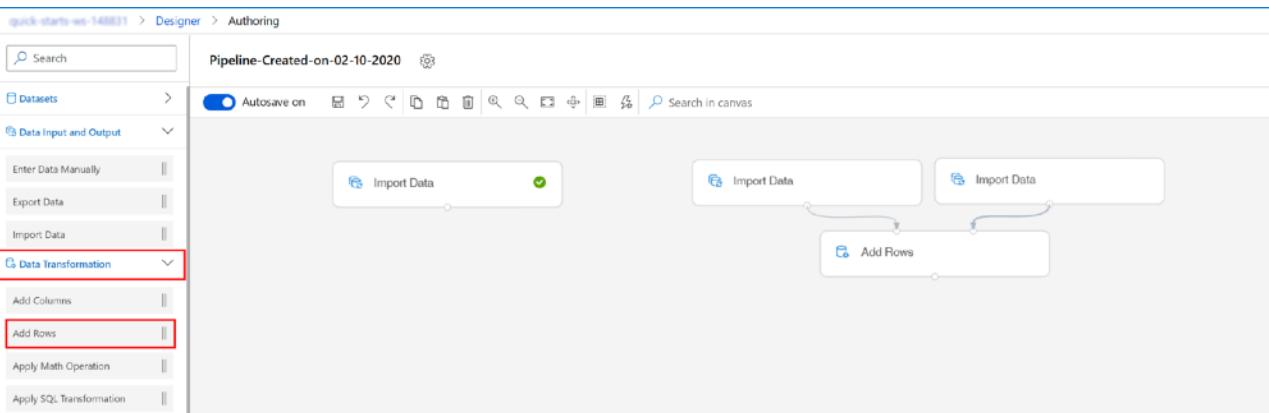
Task 1: Append rows from two additional data sources

- Select **Data Input and Output** section in the left navigation. Next, drag and drop two **Import Data** modules on to the canvas as demonstrated in the first exercise and fill in the Web URLs as follows:
 - for the first one, **Data source URL** : <https://introtomlsampled.blob.core.windows.net/data/crime-data/crime-spring.csv>
 - for the second one, **Data source URL** : <https://introtomlsampled.blob.core.windows.net/data/crime-data/crime-winter.csv>

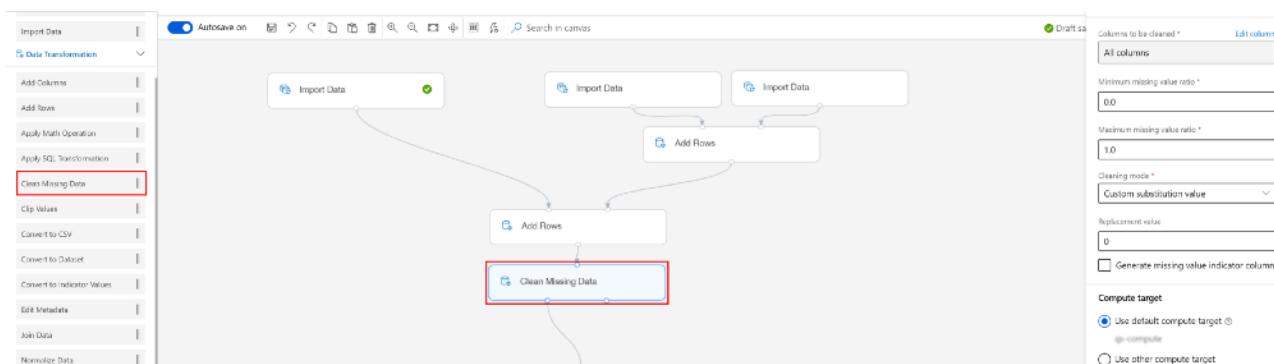
2. For each of the **three Import Data** modules, select **Preview schema** and ensure that the data type for **FBI Code** and **Location** is of type **String** and then select **Save**.



3. Select the **Data Transformation** section in the left navigation. Drag and drop the **Add rows** module and connect it to the above added Import data modules.

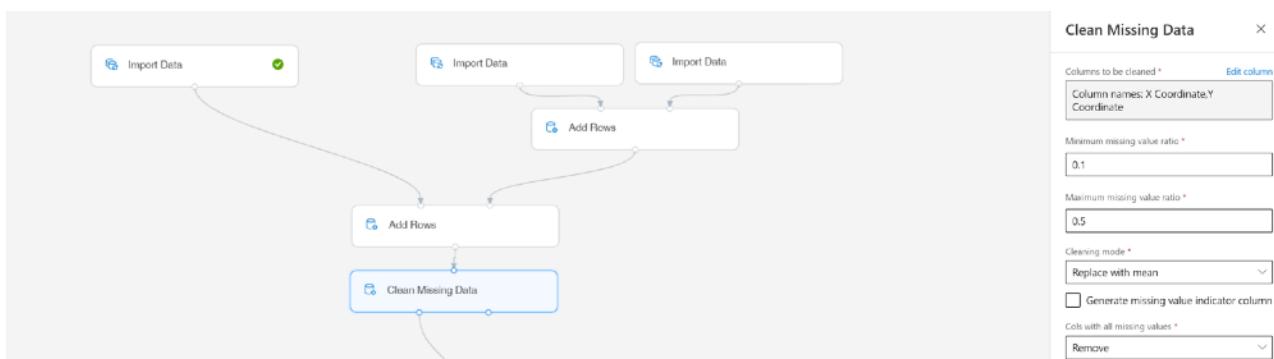


4. Repeat the same step and add a second **Add rows** module that connects the output from the first **Import data** module to the output of the first **Add rows** module.



Task 2: Clean missing values

1. Drag the **Clean Missing Data** module from the **Data Transformation** section in the left navigation.



2. Select **Edit column** in the right pane to configure the list of columns to be cleaned. Select **Column names** from the available include options and type the name of the columns you intend to clean at this step: **X Coordinate** and **Y Coordinate**. Select **Save** to close the dialog.

Columns to be cleaned

X

Select columns

With rules

By name

Allow duplicates and preserve column order in selection

Include

Column names

X Coordinate Y Coordinate

+

Save

Cancel

3. Set the **Minimum missing value ratio** to **0.1** and the **Maximum missing value ratio** to **0.5**. Select **Replace with mean** in the **Cleaning mode** field.

Clean Missing Data X

Parameters Outputs Logs Details

Columns to be cleaned * [Edit column](#)

Column names: X Coordinate,Y Coordinate

Minimum missing value ratio *

0.1

Maximum missing value ratio *

0.5

Cleaning mode *

Replace with mean

Generate missing value indicator column

Cols with all missing values *

Remove

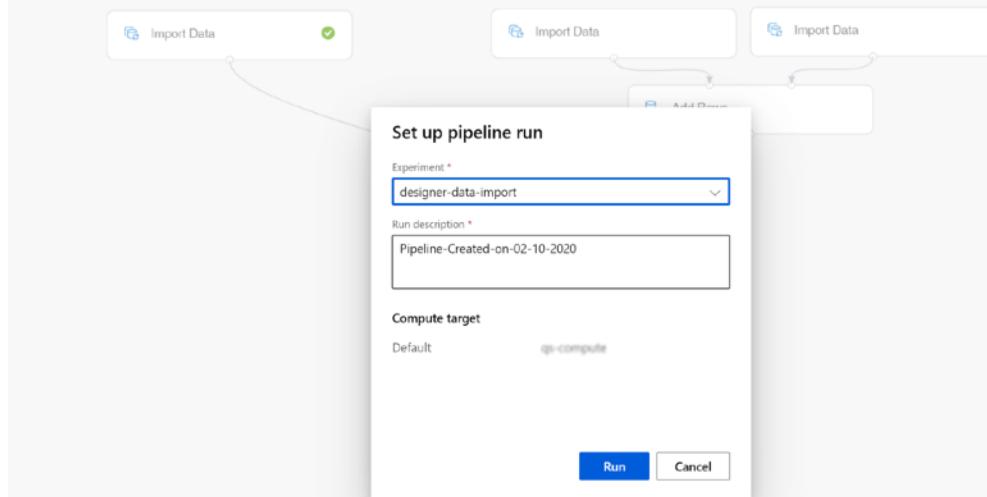
Compute target

Use default compute target ⓘ-compute

Use other compute target

Task 3: Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run** editor.
2. In the **Setup pipeline run** editor, select **Select existing, designer-data-import** for **Experiment**, and then select **Submit**.

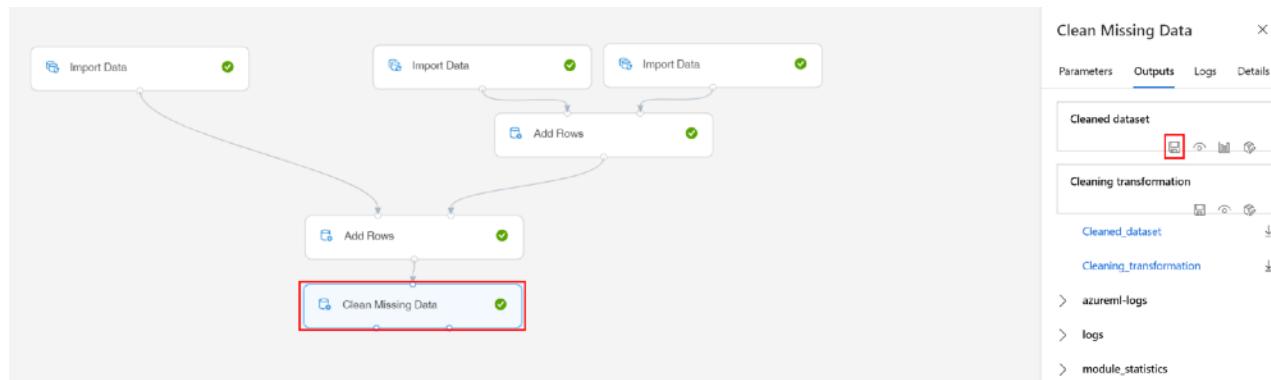


Please note that the button name in the UI is changed from **Run** to **Submit**.

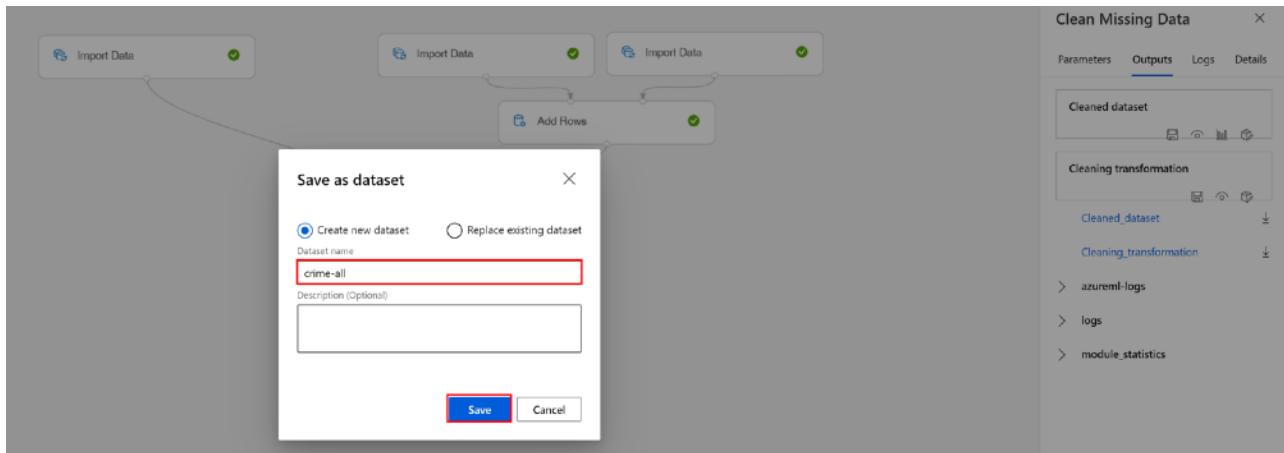
3. Wait for pipeline run to complete. It will take around **8 minutes** to complete the run.

Task 4: Save the clean dataset

1. Select the **Clean missing data** module you created on the canvas and then select **Outputs + logs** on the right pane. Click on the **Save** icon under the **Cleaned dataset** section to open the **Save as dataset** dialog.



2. Check the option to create a new dataset and enter **crime-all** in the dataset name field. Select **Save** to close the dialog.



- From the left navigation, select **Datasets**. This will open the **Registered datasets** page. See your registered dataset among the other datasets you used during this lesson.

Name	Version	Created on	Modified on	Properties	Create...	Tags
crime-all	1	Feb 25, 2020 6:31 AM	Feb 25, 2020 6:31 AM	File	ODL...	azureml.Desi...
nyc-taxi-sample-dataset	2	Feb 25, 2020 6:27 AM	Feb 25, 2020 6:29 AM	Tabular	ODL...	

Next Steps

Congratulations! You completed a few basic steps involved in the data explore and transform process, using the prebuilt modules you can find in the visual editor provided by Azure Machine Learning Studio. 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.

Lab 3 - Create and version a dataset

Overview

To access your data in your storage account, Azure Machine Learning offers datastores and datasets. Create an Azure Machine Learning datasets to interact with data in your datastores and package your data into a consumable object for machine learning tasks. Register the dataset to your workspace to share and reuse it across different experiments without data ingestion complexities.

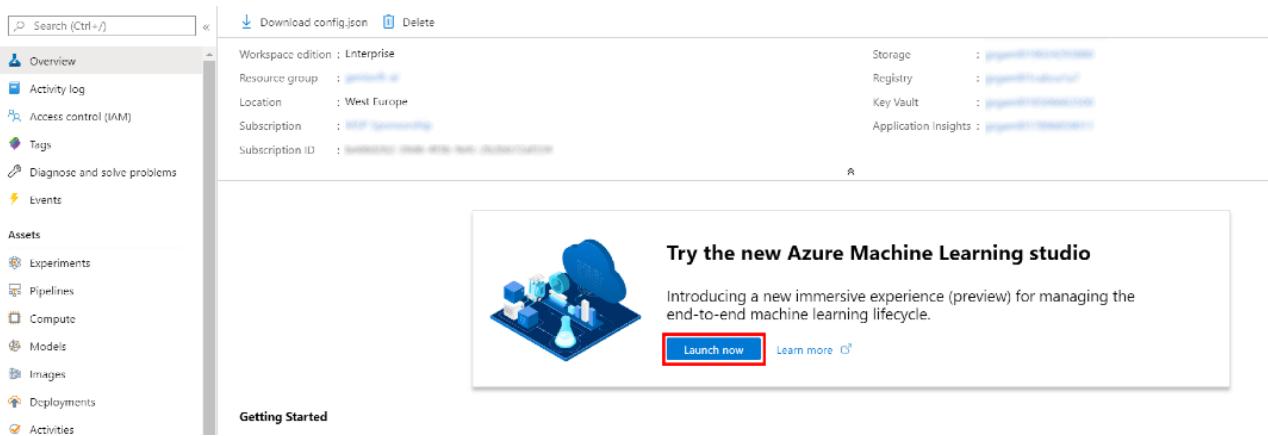
Datasets can be created from local files, public urls, Azure Open Datasets, or specific file(s) in your datastores. To create a dataset from an in memory pandas dataframe, write the data to a local file, like a csv, and create your dataset from that file. Datasets aren't copies of your data, but are references that point to the data in your storage service, so no extra storage cost is incurred.

In this lab, we are using a subset of NYC Taxi & Limousine Commission - green taxi trip records available from [Azure Open Datasets](#) to show how you can register and version a Dataset using the AML designer interface. In the first exercises we use a modified version of the original CSV file, which includes collected records for five months (January till May). The second exercise demonstrates how we can create a new version of the initial dataset when new data is collected (in this case, we included records collected in June in the CSV file).

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset from web file

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

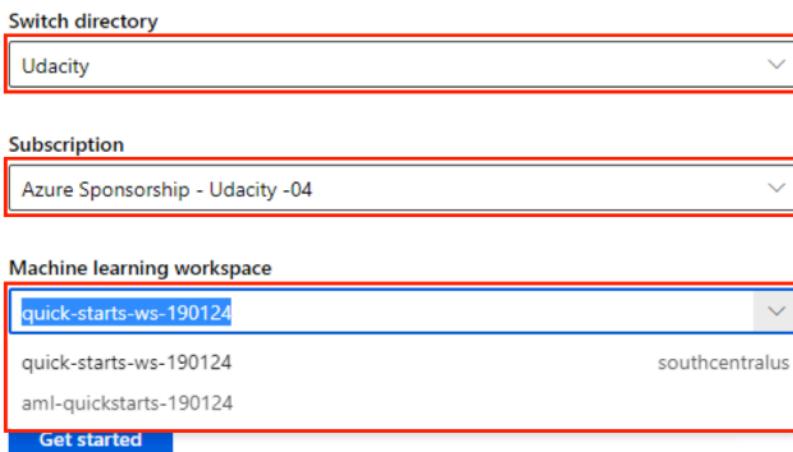


The screenshot shows the Azure Machine Learning studio landing page. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets (Experiments, Pipelines, Compute, Models, Images, Deployments, Activities), and Getting Started. The main area displays workspace details: Workspace edition: Enterprise; Resource group: [redacted]; Location: West Europe; Subscription: [redacted]; Subscription ID: [redacted]. Below these are storage, Registry, Key Vault, and Application Insights links. A central callout box says 'Try the new Azure Machine Learning studio', featuring a 3D model of a circuit board with data points. It includes a 'Launch now' button (which is highlighted with a red box) and a 'Learn more' link.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



The screenshot shows the 'Welcome to the studio!' setup screen. It has three dropdown menus:

- Switch directory:** Set to 'Udacity'.
- Subscription:** Set to 'Azure Sponsorship - Udacity -04'.
- Machine learning workspace:** Shows a list of workspaces:
 - 'quick-starts-ws-190124' (selected, highlighted with a blue border)
 - 'quick-starts-ws-190124'
 - 'aml-quickstarts-190124'A 'Get started' button is located at the bottom of this section.

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**.

4. From the studio, select **Datasets**, + **Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

5. Provide the following information and then select **Next**:

1. Web URL: <https://introtomlsampleddata.blob.core.windows.net/data/nyc-taxi/nyc-taxi-sample-data-5months.csv>
2. Name: nyc-taxi-sample-dataset

Create dataset from web files X

Basic info
Settings and preview
Schema
Confirm details

Basic info

Web URL *

Name *

Dataset version

Dataset type *

Description

Skip data validation

Next
Back
Cancel

Task 2: Preview Dataset

1. On the Settings and preview panel, set the **Column headers** drop down to **All files have same headers**.
2. Scroll the data preview to right to observe the target column: **totalAmount**. After you are done reviewing the data, select **Next**

Create dataset from web files X

Basic info

Settings and preview

Schema

Confirm details

Settings and preview

These settings were automatically detected. Please verify that the selections were made correctly or update.

File format: Delimited

Delimiter: Comma Example: Field1,Field2,Field3

Encoding: UTF-8

Column headers: All files have same headers

Skip rows: None

0.0 snowDepth	0.0 precipTime	0.0 precipDepth	0.0 temperature	0.0 totalAmount
29.05882353	24	3	6.185714286	44.3
0	6	0	4.571929825	44.8
0	1	0	4.384090909	18.96
29.05882353	24	3	6.185714286	16.3
0	1	0	3.846428571	5.3
0	6	0	0.159459459	16.3

Back Next Cancel

Task 3: Select Columns

1. Select columns from the dataset to include as part of your training data. Leave the default selections and select **Next**

Create dataset from web files X

Basic info

Settings and preview

Schema

Confirm details

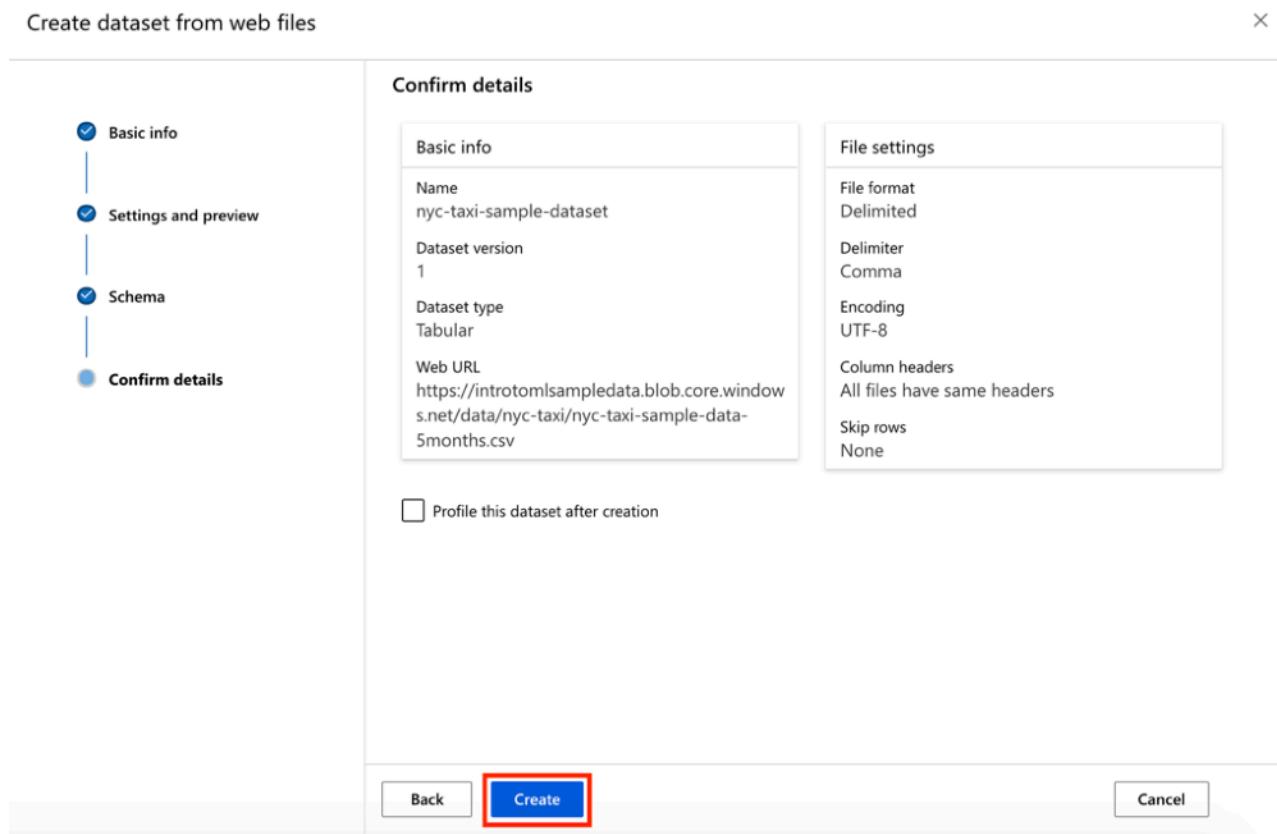
Schema

Include	Column name	Properties	Type
<input checked="" type="checkbox"/>	Path	Not applicable to selected columns	String
<input checked="" type="checkbox"/>	vendorID	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	passengerCount	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	tripDistance	Not applicable to selected columns	Decimal
<input checked="" type="checkbox"/>	hour_of_day	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	day_of_week	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	day_of_month	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	month_num	Not applicable to selected columns	Integer
<input checked="" type="checkbox"/>	normalizeHolidayName	Not applicable to selected columns	String
<input checked="" type="checkbox"/>	isPaidTimeOff	Not applicable to selected columns	Boolean

Back Next Cancel

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**



Exercise 2: Create a version of the existing Dataset

Task 1: Register new dataset version

1. From the [Azure Machine Learning studio](#), select **Datasets** and select the [nyc-taxi-sample-dataset](#) dataset created in the first exercise. This will open the [Dataset details](#) page.
2. Select **New version, From web files** to open the same [Create dataset from web files](#) dialog you already entered in the first exercise.

The screenshot shows the Azure Machine Learning studio interface. On the left, there's a sidebar with various navigation options like New, Home, Author, Notebooks, Automated ML, Designer, Datasets (which is selected and highlighted with a red box), Experiments, Pipelines, Models, Endpoints, Compute, Datastores, and Data Labeling. The main area displays a dataset named "nyc-taxi-sample-dataset" (also highlighted with a red box). A dropdown menu for "Version 1 (latest)" is open, showing options: Refresh, Generate profile, Unregister, New version (highlighted with a red box), From local files, From datastore, From web files (highlighted with a red box), and From Open Datasets. The dataset details include attributes like Properties (Tabular), Description (empty), Created by (redacted), Web Url (https://introtomlsampledatal.blob.core.windows.net/data/nyc-taxi/nyc-taxi-sample-data-5months.csv), Profile (No profile generated), Current version (1), Latest version (1), Created time (Jun 16, 2020 3:57 PM), and Modified time (Jun 16, 2020 3:57 PM).

3. This time, the **Name** and **Dataset version** fields are already filled in for you. Provide the following information and select **Next** to move on to the next step:
 1. Web URL: <https://introtomlsampledatal.blob.core.windows.net/data/nyc-taxi/nyc-taxi-sample-data-6months.csv>

Create dataset from web files

Basic info

Web URL *

Name * Dataset version

Dataset type *

Description

Skip data validation

Back **Next** Cancel

4. Select **All files have the same headers** in the **Column headers** drop-down and move on to the schema selection step.
5. On the **Schema** page, let's suppose you decided to exclude some columns from your dataset. Exclude columns: **snowDepth**, **precipTime**, **precipDepth**. Select **Next** to move on to the final step.

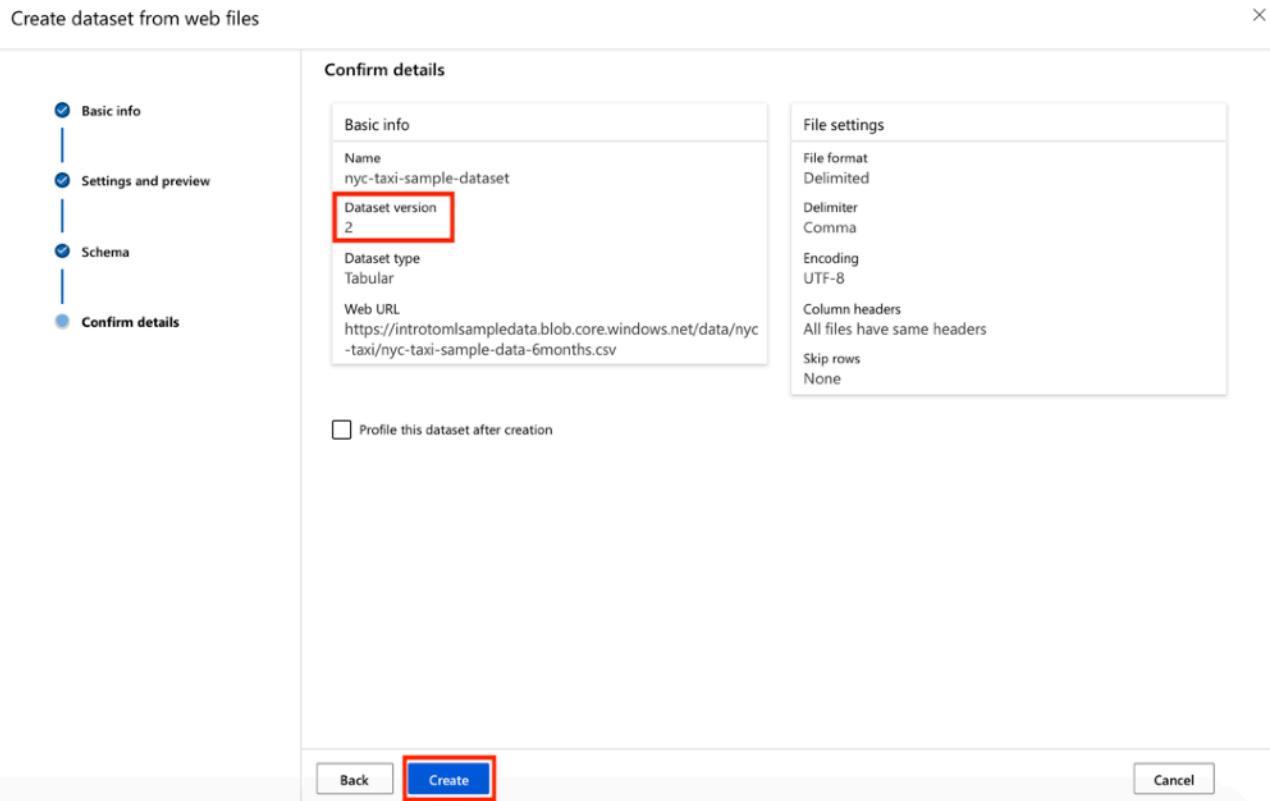
Create dataset from web files

Schema

Include	Column name	Properties	Type	Format settings
<input checked="" type="checkbox"/>	hour_of_day	Not applicable to select...	Integer	15, 13, 23
<input checked="" type="checkbox"/>	day_of_week	Not applicable to select...	Integer	2, 4, 4
<input checked="" type="checkbox"/>	day_of_month	Not applicable to select...	Integer	27, 15, 8
<input checked="" type="checkbox"/>	month_num	Not applicable to select...	Integer	1, 1, 1
<input checked="" type="checkbox"/>	normalizeHolidayName	Not applicable to select...	String	None, None, Non
<input checked="" type="checkbox"/>	isPaidTimeOff	Not applicable to select...	Boolean	false, false, false
<input checked="" type="checkbox"/>	snowDepth	Not applicable to select...	Decimal	29.058823529411
<input checked="" type="checkbox"/>	precipTime	Not applicable to select...	Decimal	24, 6, 1
<input checked="" type="checkbox"/>	precipDepth	Not applicable to select...	Decimal	3, 0, 0
<input checked="" type="checkbox"/>	temperature	Not applicable to select...	Decimal	6.1857142857142
<input checked="" type="checkbox"/>	totalAmount	Not applicable to select...	Decimal	44.3, 44.8, 18.96

Back **Next** Cancel

6. Notice the **Dataset version** value in the basic info section. Select **Create** to close the new version confirmation page.



Task 2: Review both versions of the dataset

1. Back to the **Datasets** page, in the **Registered datasets** list, notice the version value for the **nyc-taxi-sample-dataset** dataset.

Name	Version	Created on	Modified on	Properties
nyc-taxi-sample-dataset	2	Feb 21, 2020 1:15 PM	Feb 21, 2020 1:23 PM	Tabular

2. Select the **nyc-taxi-sample-dataset** dataset link to open the dataset details page, where **Version 2(latest)** is automatically selected. Go to the **Explore** section to observe the structure and content of the new version. Notice the columns and rows structure in the dataset preview pane:

- **Number of columns:** 11

- **Number of rows:** 10000
- Scroll right to check that the three excluded columns are missing (**snowDepth**, **precipTime**, **precipDepth**)

nyc-taxi-sample-dataset Version 2 (latest) 

Details Consume **Explore** Models

 Refresh  Generate profile  Unregister  New version 

 Profile: This is the quick profile generated by sampled data. Please generate a profile from the action bar to view the full profile. 

Preview Profile

	Number of columns: 11	Number of rows: 50 (of 10000)								
	passengerCount	tripDistance	hour of day	day of week	day of month	month num	normalizeHoli...	isPaidTimeOff	temperature	totalAmount
1	9.4	15	2	27	1	None	false	6.18571428571429	44.3	
5	14.75	13	4	15	1	None	false	4.571929824561403	44.8	
1	3.35	23	1	8	1	None	false	4.3840909090909013	18.06	
1	3.13	18	2	27	1	None	false	6.18571428571429	16.3	
1	0.47	17	6	3	1	None	false	3.846428571428569	5.3	
1	3.07	9	1	12	1	None	false	0.1594594594594597	16.3	
1	0.92	23	4	22	1	None	false	-2.999107142857142	8.97	
1	1.9	12	4	8	1	None	false	4.384090909090913	11.8	
1	0.77	0	1	19	1	None	false	5.393749999999998	7.3	
1	2.35	2	6	10	1	None	false	10.943654822335034	14.16	
1	8.3	18	3	21	1	None	false	-0.0400000000000000...	34.3	
2	4.28	18	0	18	1	Martin Luther King, J...	true	2.3351145038167944	18.06	
1	10.77	2	2	27	1	None	false	6.18571428571429	31.3	
1	1.75	17	3	14	1	None	false	-1.9500000000000008	14.3	
1	3.75	2	4	1	1	New Year's Day	true	5.197345132743359	19.3	
1	5.79	14	6	3	1	None	false	3.846428571428569	33.55	
5	1.06	19	4	29	1	None	false	3.1651785714285696	8.3	
1	5.7	11	2	13	1	None	false	2.006875	20.1	
1	3.26	10	3	14	1	None	false	1.9500000000000008	23.34	

3. Select **Version 1** from the drop-down near the dataset name title and notice the changing values for:

- **Number of columns:** 14 (since the previous version still contains the three excluded columns)
- **Number of rows:** 9776 (since the previous version contains only data for 5 months)

nyc-taxi-sample-dataset Version 1 

Details Consume **Explore** Models

 Refresh  Generate profile  Unregister  New version 

 Profile: This is the quick profile generated by sampled data. Please generate a profile from the action bar to view the full profile. 

Preview Profile

	Number of columns: 14	Number of rows: 50 (of 9776)									
	our_of_day	day_of_week	day_of_month	month_num	normalizeHoli...	isPaidTimeOff	00_snowDepth	00_precipTime	00_precipDepth	00_temperature	00_tr
1	2	27	1	None	false	29.05882353	24	3	6.185714286	44.3	
4	4	15	1	None	false	0	6	0	4.571929825	44.8	
4	4	8	1	None	false	0	1	0	4.384090909	18.06	
2	2	27	1	None	false	29.05882353	24	3	6.185714286	16.3	
6	6	3	1	None	false	0	1	0	3.846428571	5.3	
1	1	12	1	None	false	0	6	0	0.159459459	16.3	
4	4	22	1	None	false	0	1	0	-2.999107143	8.97	
4	4	8	1	None	false	0	1	0	4.384090909	11.8	
1	1	19	1	None	false	0	1	0	-5.39375	7.3	
6	6	10	1	None	false	0	24	254	10.94365482	14.16	
3	3	21	1	None	false	0	1	0	-0.04	34.3	
0	0	18	1	Martin Luther King, J...	true	3	24	13	-2.335114504	18.96	
2	2	27	1	None	false	29.05882353	24	3	6.185714286	31.3	
3	3	14	1	None	false	0	6	0	-1.95	14.3	
4	4	1	1	New Year's Day	true	0	1	0	5.197345133	19.3	
6	6	3	1	None	false	0	1	0	3.846428571	33.55	
4	4	29	1	None	false	15.64705882	6	0	3.365178571	8.3	
2	2	13	1	None	false	0	6	0	-2.06875	29.1	
3	3	14	1	None	false	0	6	0	-1.95	23.34	

Next Steps

Congratulations! You have now explored a first simple scenario for dataset versioning using the Azure Machine Learning studio. You found out how you can create and version a simple dataset when new training data is available. 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.

Lab 4 - Engineer and select features

Overview

This lab demonstrates the feature engineering process for building a regression model using bike rental demand prediction as an example. In machine learning predictions, effective feature engineering will lead to a more accurate model. We will use the Bike Rental UCI dataset as the input raw data for this experiment. This dataset is based on real data from the Capital Bikeshare company, which operates a bike rental network in Washington DC in the United States. The dataset contains 17,379 rows and 17 columns, each row representing the number of bike rentals within a specific hour of a day in the years 2011 or 2012. Weather conditions (such as temperature, humidity, and wind speed) were included in this raw feature set, and the dates were categorized as holiday vs. weekday etc.

The field to predict is `cnt` which contains a count value ranging from 1 to 977, representing the number of bike rentals within a specific hour. Our main goal is to construct effective features in the training data, so we build two models using the same algorithm, but with two different datasets. Using the Split Data module in the visual designer, we split the input data in such a way that the training data contains records for the year 2011, and the testing data, records for 2012. Both datasets have the same raw data at the origin, but we added different additional features to each training set:

- Set A = weather + holiday + weekday + weekend features for the predicted day
- Set B = number of bikes that were rented in each of the previous 12 hours

We are building two training datasets by combining the feature set as follows:

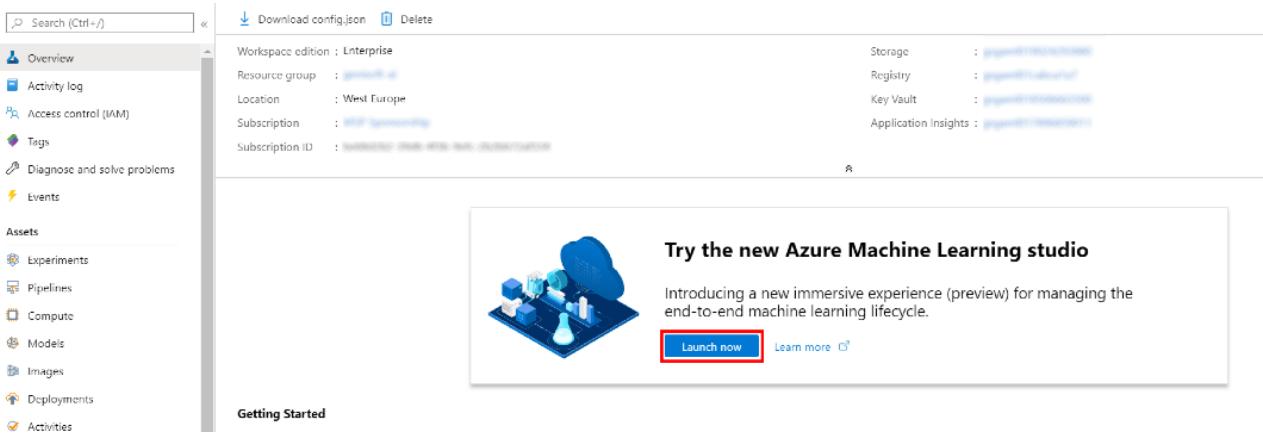
- Training set 1: feature set A only
- Training set 2: feature sets A+B

For the model, we are using regression because the number of rentals (the label column) contains continuous real numbers. As the algorithm for the experiment, we will be using the Boosted Decision Tree Regression.

Exercise 1: Data pre-processing using the Pipeline Authoring Editor

Task 1: Upload Dataset

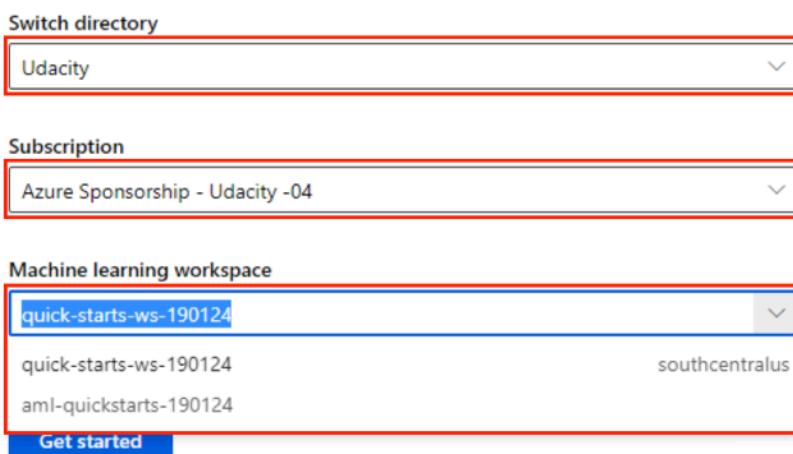
1. In [Azure portal](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



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**.

4. From the studio, select **Datasets, + Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the Azure Machine Learning Studio interface. On the left, there's a sidebar with various options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets' (which is selected and highlighted with a red box), 'Experiments', 'Pipelines', 'Models', 'Endpoints', 'Manage', 'Compute', 'Datastores', and 'Data Labeling'. The main area is titled 'Datasets' and shows 'Registered datasets'. It has a sub-menu with options: '+ Create dataset', 'Refresh', 'Unregister', 'From local files', 'From datastore', 'From web files' (which is also highlighted with a red box), and 'From Open Datasets'. There's a table with columns 'Version' and 'Created on'.

5. In the Web URL field provide the following URL for the training data file:
<https://introtomlsampleddata.blob.core.windows.net/data/bike-rental/bike-rental-hour.csv>
6. Provide **Bike Rental Hourly** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows a 'Create dataset from web files' dialog. On the left, there's a navigation pane with tabs: 'Basic info' (selected and highlighted with a blue circle), 'Settings and preview', 'Schema', and 'Confirm details'. The main area is titled 'Basic info' and contains the following fields:

- 'Web URL *' with the value <https://introtomlsampleddata.blob.core.windows.net/data/bike-rental/bike-rental-hour.csv>, which is highlighted with a red box.
- 'Name *' with the value 'Bike Rental Hourly', which is also highlighted with a red box.
- 'Dataset type *' with the value 'Tabular'.
- 'Description' with the value 'Dataset description'.

7. Select the option to **Use headers from the first file** in the **Settings and preview** dialog and then select **Next**, **Next** and **Create** to confirm all details in registering the dataset.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there's a navigation sidebar with 'Basic info' (selected), 'Settings and preview' (highlighted with a blue circle), 'Schema', and 'Confirm details'. The main area is titled 'Settings and preview' with a sub-section 'File format' set to 'Delimited'. Under 'Column headers', the dropdown 'Use headers from the first file' is selected and highlighted with a red box. Below this is a 'Skip rows' dropdown set to 'None'. A large table follows, showing 11 rows of data with columns: instant, dteday, season, yr, mnth, hr, holiday, weekday, workingday, weathersit, tempr, atemp, hum, windspeed, casual, registered, and cnt. At the bottom are three buttons: 'Back', 'Next' (highlighted with a red box), and 'Cancel'.

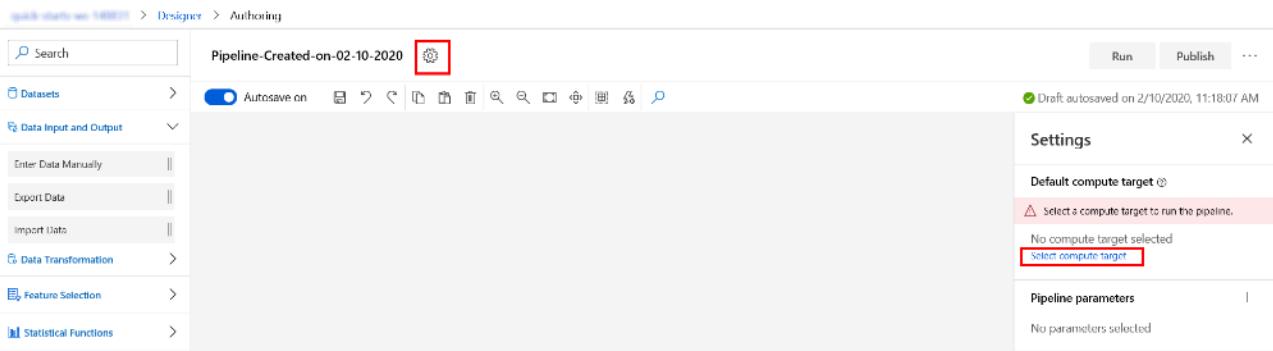
Task 2: Open Pipeline Authoring Editor

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

The screenshot shows the 'Designer' page. The left sidebar has sections for 'Author' (selected), 'Notebooks', 'Automated ML', 'Designer' (highlighted with a red box), 'Assets', 'Datasets', 'Experiments', 'Pipelines', 'Models', 'Endpoints', 'Manage', 'Compute', 'Datastores', and 'Pipelines'. The main area has a 'New pipeline' section with a large '+' button (highlighted with a red box). Below are four sample modules: 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', and 'Sample 4 with cust...'. At the bottom is a 'Pipelines' section with 'Pipeline drafts' and 'Pipeline runs' tabs, a 'Refresh' button, a 'Delete' button, and a search bar 'Search to filter items...'.

Task 3: Setup Compute Target

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

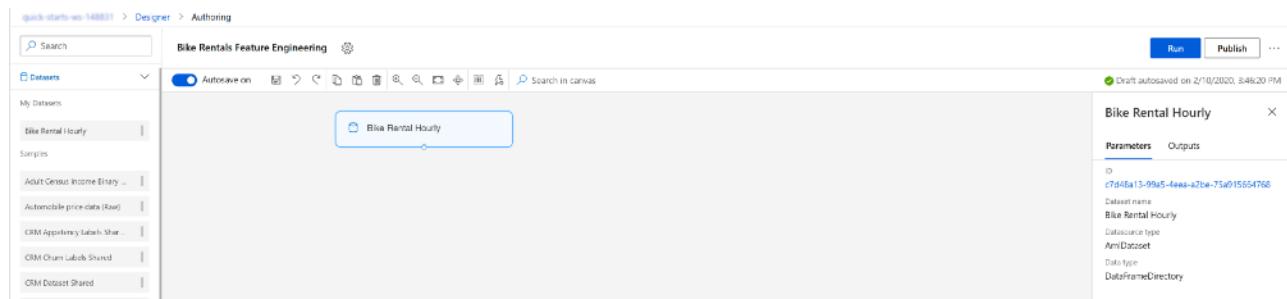


2. In the **Set up compute target** editor, select the existing compute target, choose a name for the pipeline draft: **Bike Rental Feature Engineering** and then select **Save**.

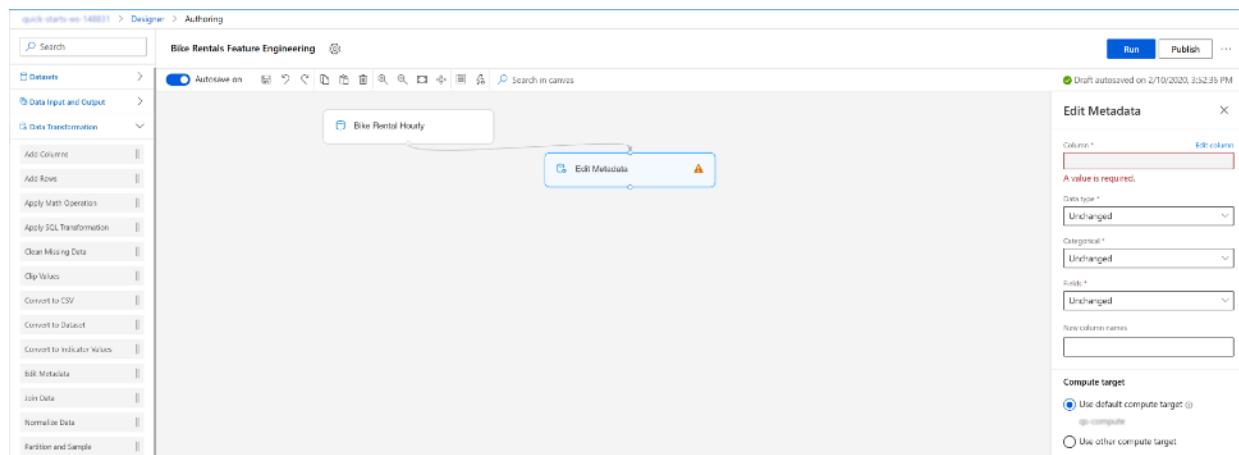
Note: If you are facing difficulties in accessing pop-up windows or buttons in the user interface, please refer to the Help section in the lab environment.

Task 4: Select columns in the dataset

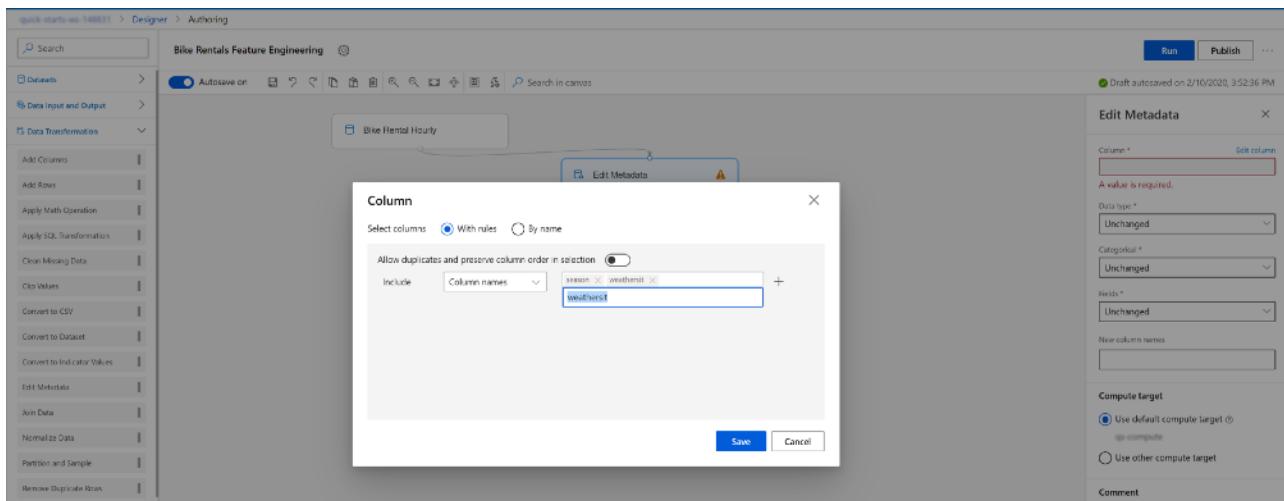
1. Drag and drop on the canvas, the available **Bike Rental Hourly** dataset under the **Datasets** category on the left navigation.



2. Under the **Data transformation** category drag and drop the **Edit metadata** module.



3. Edit the column list by selecting **Edit columns** on the right pane. Add the **season** and **weathersit** column and select **Save**.



4. Configure the Edit metadata module by selecting the **Categorical** attribute for the two columns.

Edit Metadata

Column *

Column names: season,weathersit

Column names: season,weathersit

Data type *

Unchanged

Categorical *

Categorical

Fields *

Unchanged

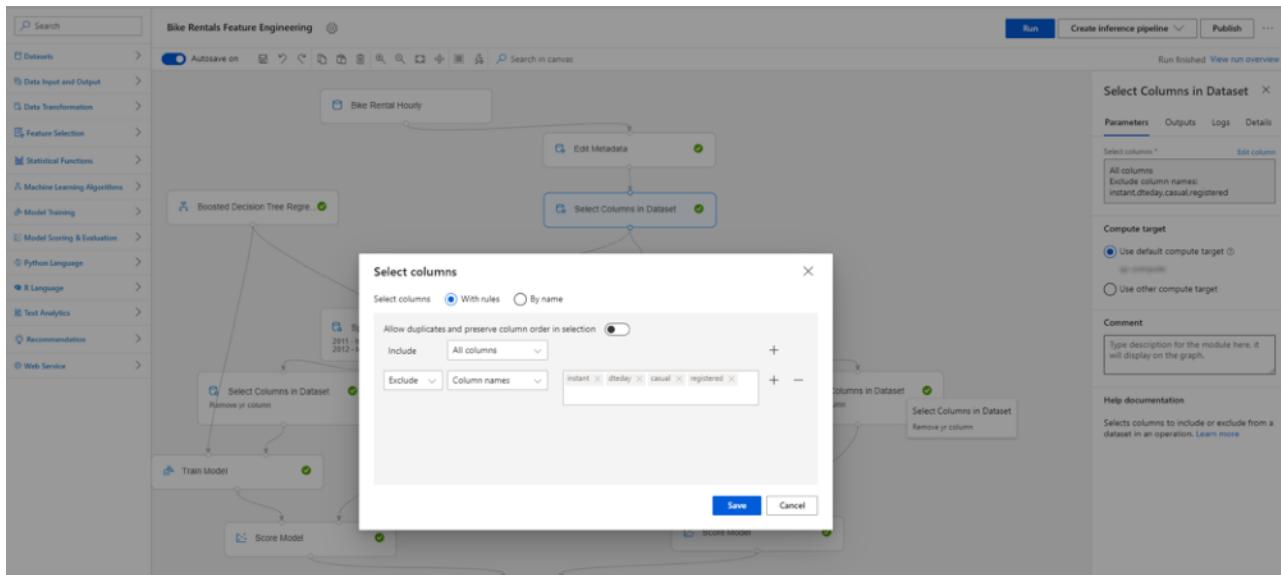
New column names

Compute target

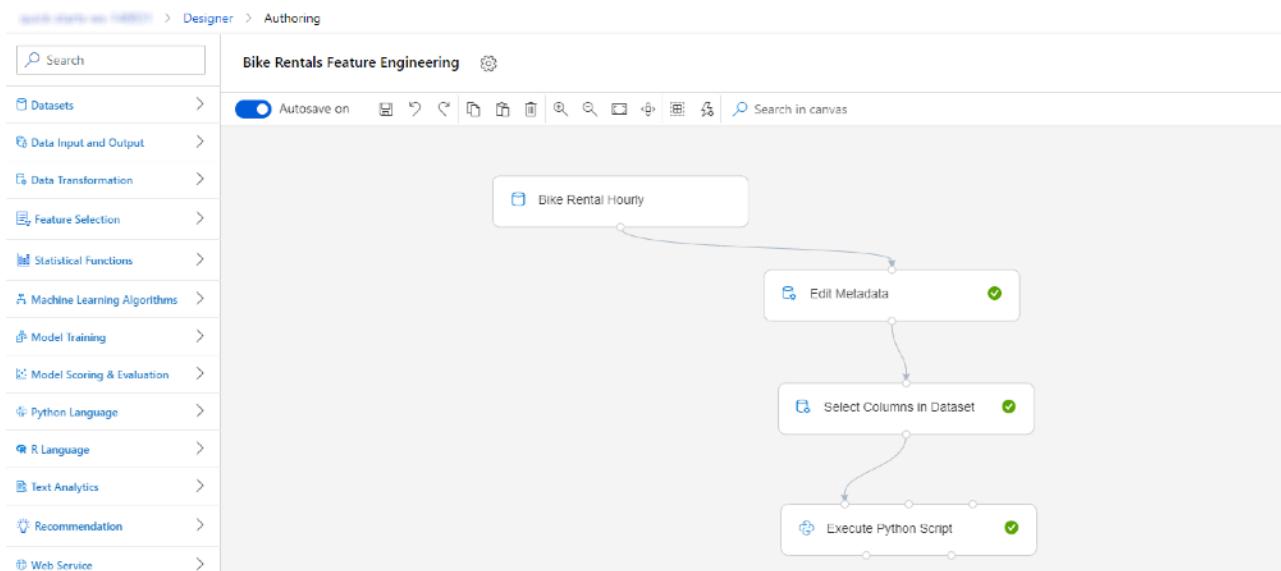
Use default compute target qs-compute

Use other compute target

5. Next, use the **Select columns in Dataset** module under the **Data transformation** category and configure it as follows:
- Include all columns
 - Exclude column names: instant, dteday, casual, registered
 - Use default compute target: aml-compute



6. Connect the output from the **Edit columns** module to the input of the **Select columns in Dataset** module.
7. Under the **Python Language** category on the left, select the **Execute Python Script** module and connect it with the **Select Columns in Dataset** module. Make sure the connector is connected to the very first input of the **Execute Python Script** module.



8. We are using the Python script to append a new set of features to the dataset: number of bikes that were rented in each of the previous 12 hours. Feature set B captures very recent demand for the bikes. This will be the B set in the described feature engineering approach.

Select **Edit code** and use the following lines of code:

```
# The script MUST contain a function named azureml_main
# which is the entry point for this module.

# imports up here can be used to
import pandas as pd
import numpy as np

# The entry point function can contain up to two input arguments:
# Param<dataframe1>: a pandas.DataFrame
# Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None, dataframe2 = None):

    # Execution logic goes here
    print(f'Input pandas.DataFrame #1: {dataframe1}')

    # If a zip file is connected to the third input port,
    # it is unzipped under "./Script Bundle". This directory is added
    # to sys.path. Therefore, if your zip file contains a Python file
    # mymodule.py you can import it using:
    # import mymodule

    for i in np.arange(1, 13):
        prev_col_name = 'cnt' if i == 1 else 'Rentals in hour -{}'.format(i-1)
        new_col_name = 'Rentals in hour -{}'.format(i)

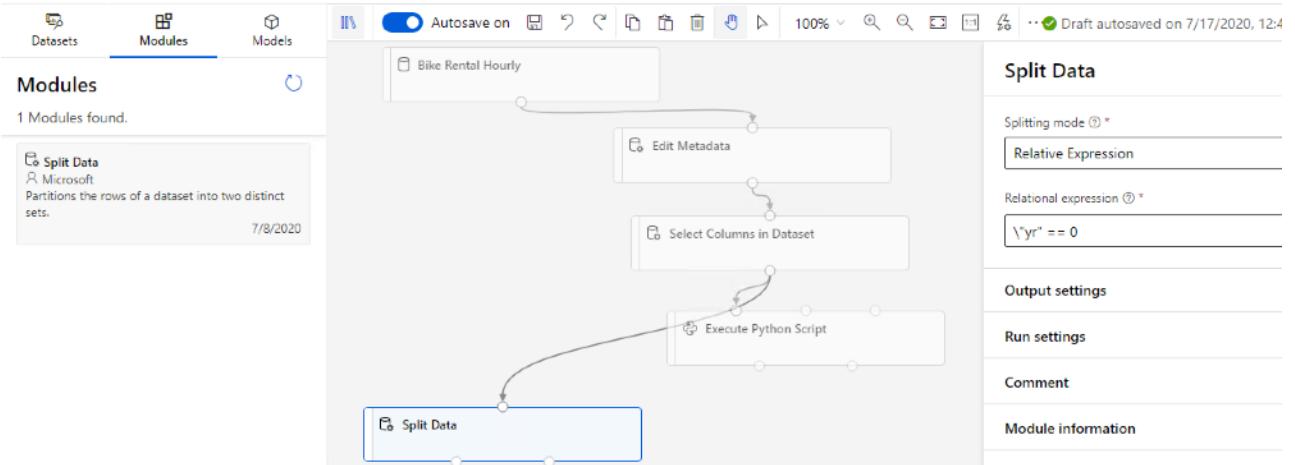
        dataframe1[new_col_name] =
dataframe1[prev_col_name].shift(1).fillna(0)

    # Return value must be of a sequence of pandas.DataFrame
    # E.g.
    # - Single return value: return dataframe1,
    # - Two return values: return dataframe1, dataframe2
    return dataframe1,
```

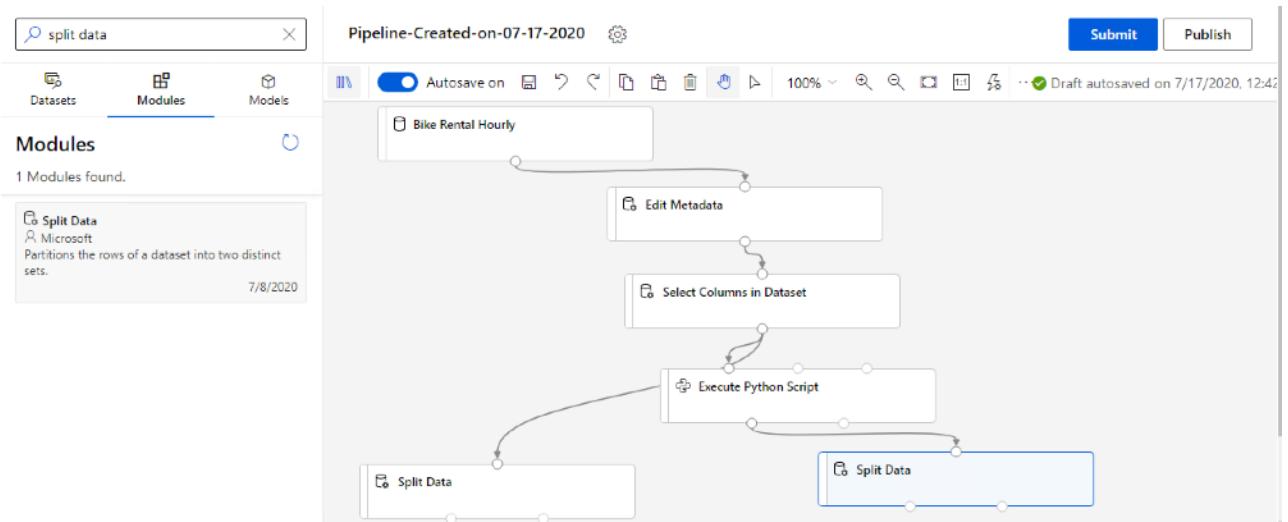
Don't worry if you do not fully understand the details of the Python code above. For now, it's enough to keep in mind that it adds 12 new columns to your dataset containing the number of bikes that were rented in each of the previous 12 hours.

Task 5: Split data into train and test datasets

1. Use the **Split Data** module under the **Data Transformation** module and connect its input with output from the **Select Columns in Dataset** module. Use the following configuration:
 - Splitting mode: **Relative Expression**
 - Relational expression: **\"yr\" == 0**

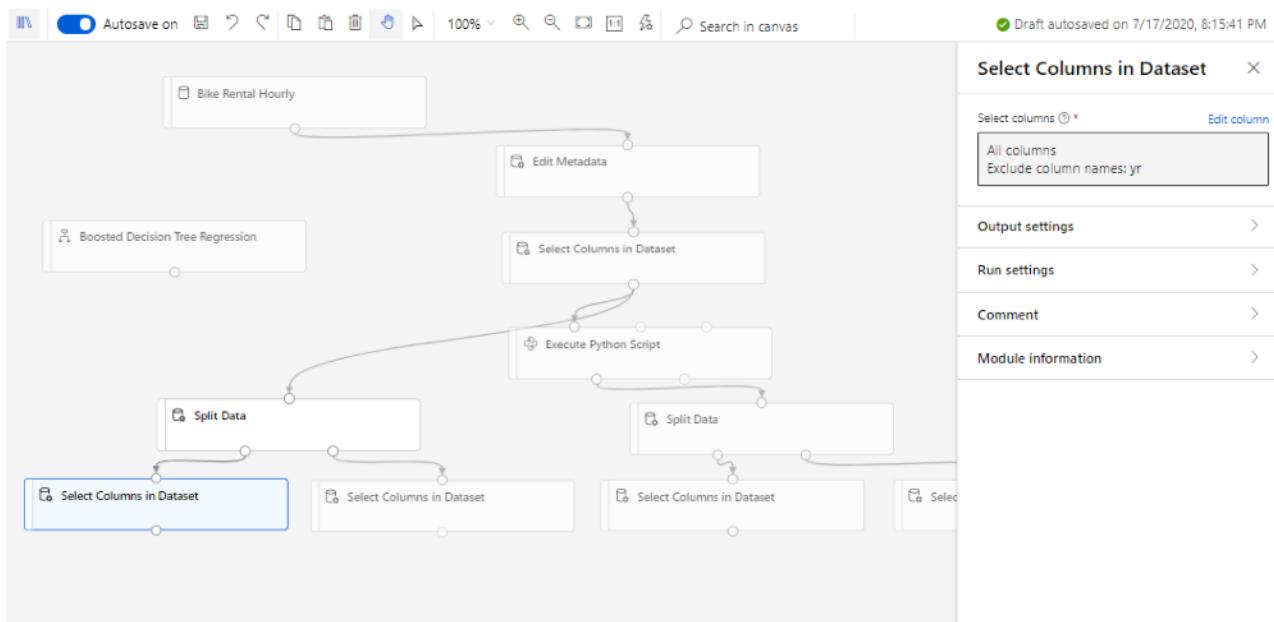


2. Select the **Split Data** module block and use the menu buttons to Copy and Paste it on the canvas. Connect the second one to the output of the Python Script execution step, which is the featured B set.

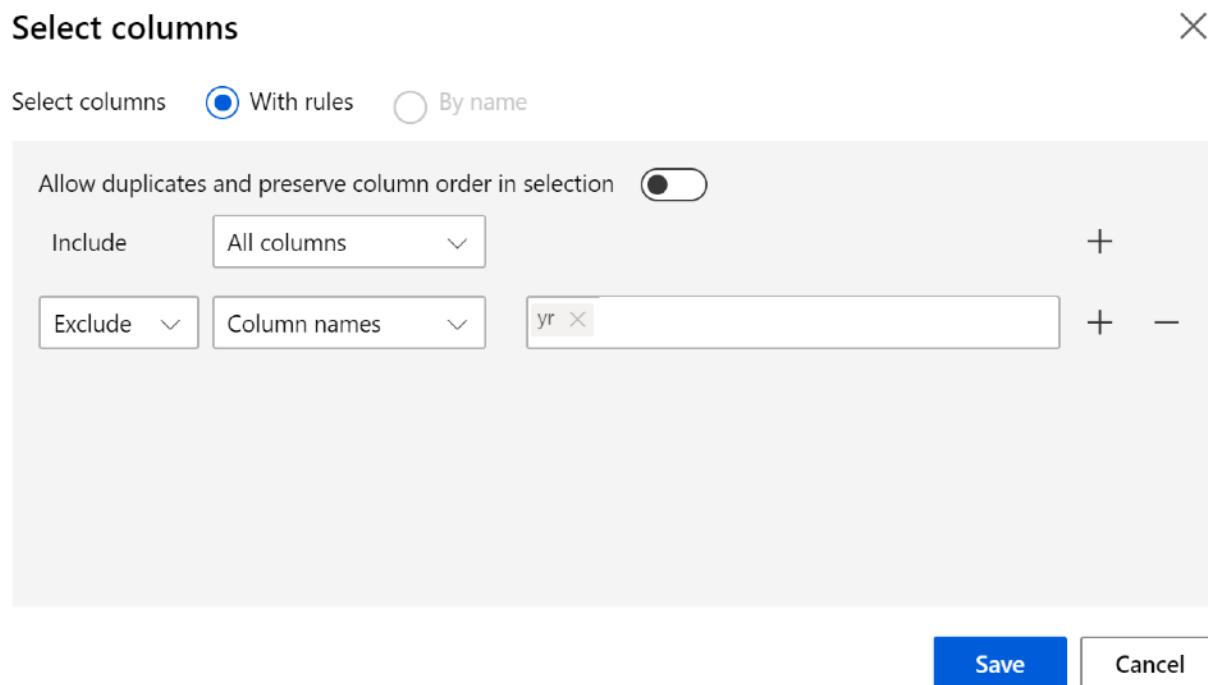


Task 6: Select columns from the test and training resulted sets

1. Next, using the **Select columns** module under the **Data transformation** category, create four identical modules to exclude the **yr** column from all the outputs: test and training sets in both branches: A and A+B.

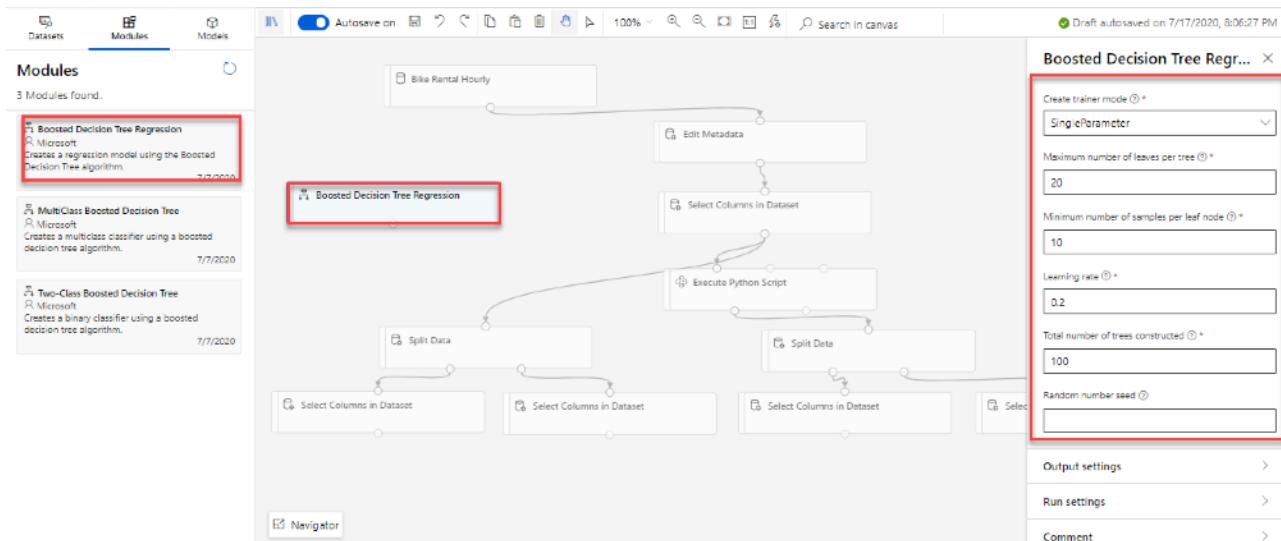


2. Use the following structure for the columns field in each module:

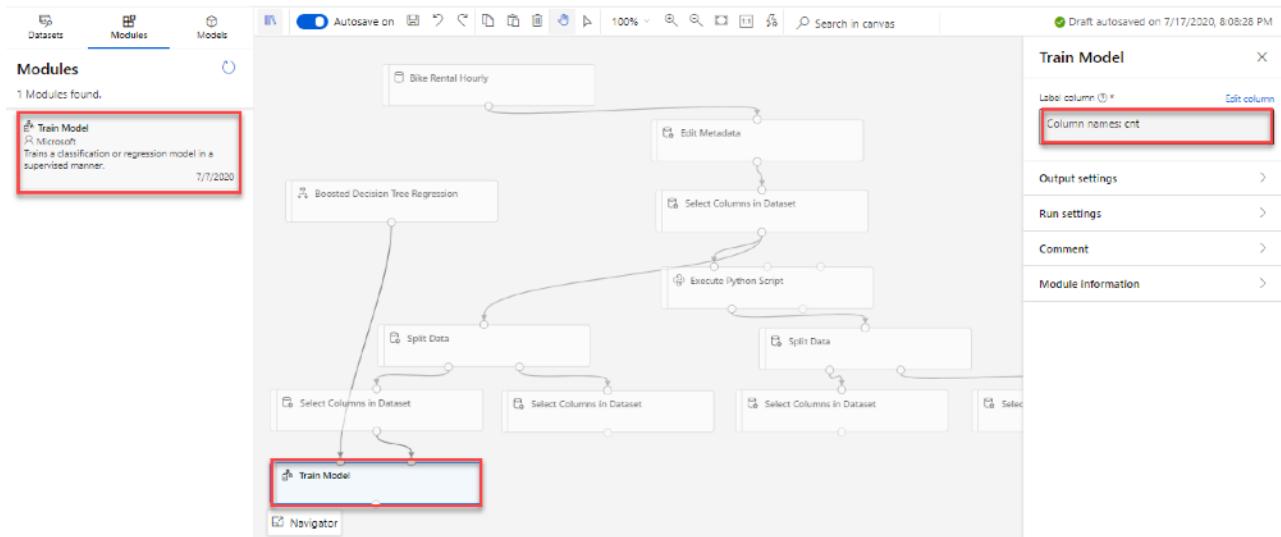


Task 7: Create the regression model

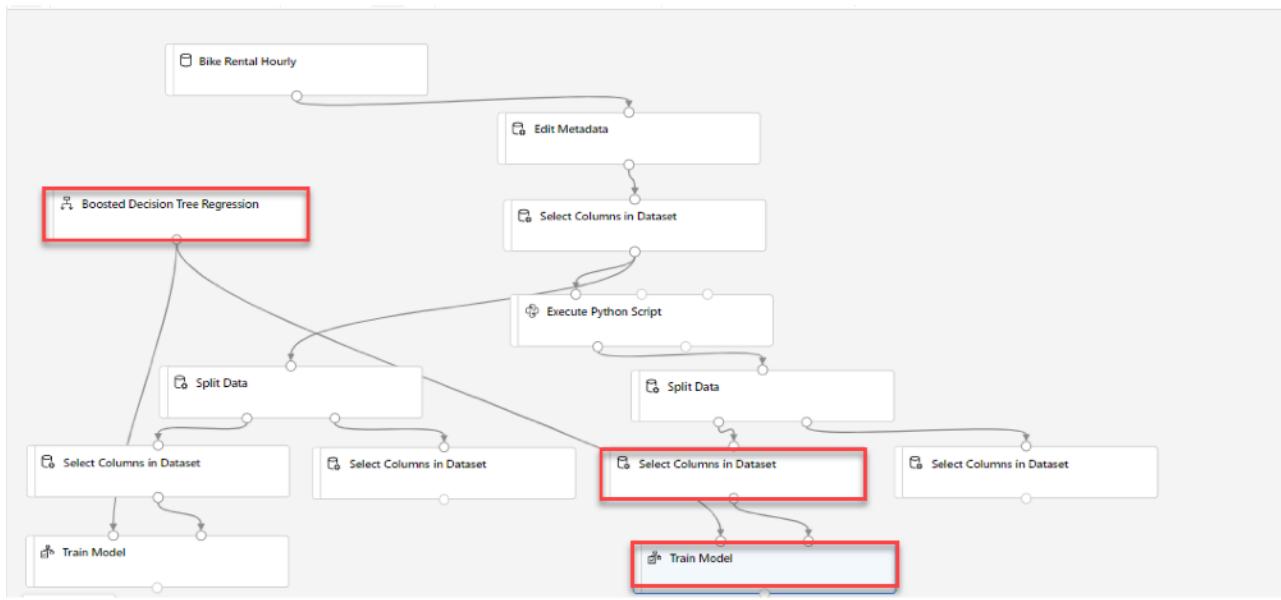
- Under the **Machine Learning Algorithms, Regression** category, select the **Boosted Decision Tree Regression** module. Drag and drop it on the canvas and use the default settings provided.



- Next, use the **Train model** module under the **Model training** category and enter the `cnt` column in the **Label column** field.
- Link the **Boosted Decision Tree Regression** module as the first input and the training dataset as the second input like in the image below.

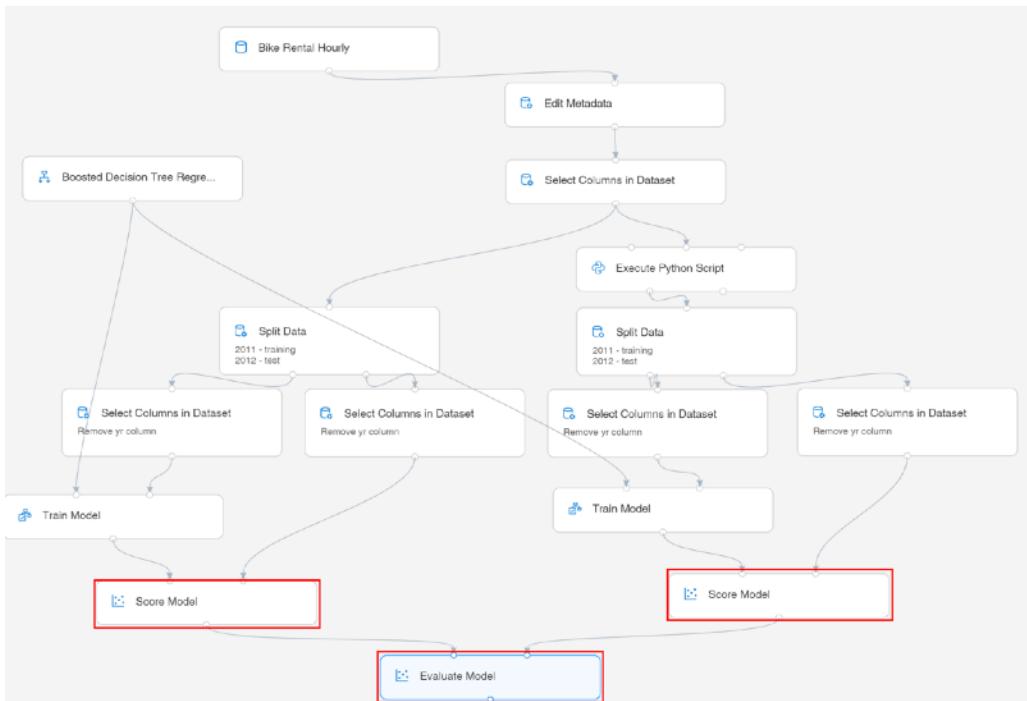


4. Use the exact same configuration on the right branch that uses the output from the Python Script.

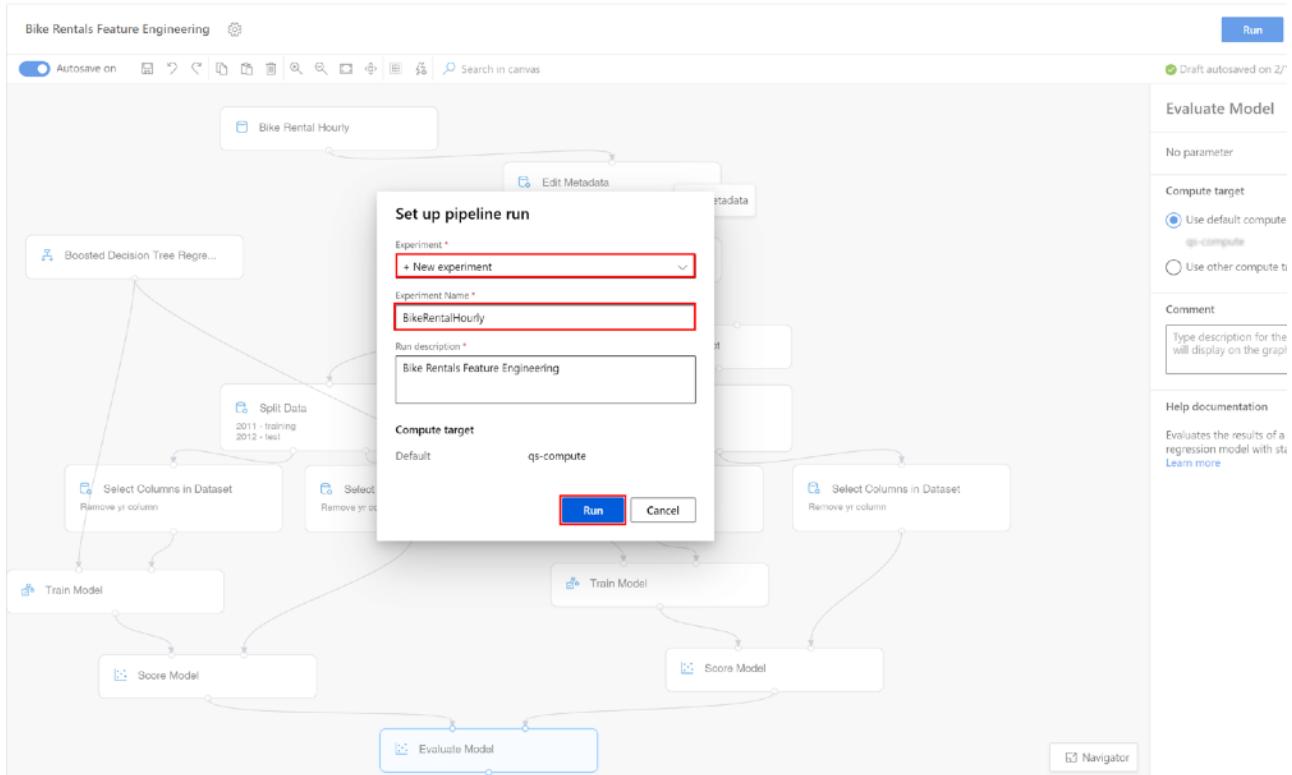


Task 8: Evaluate and score models

1. Use two **Score Model** modules (under the **Model Scoring and Evaluation** category) and link on the input the two trained models and the test datasets.
2. Drag the **Evaluate Model** module which stands in the same category, **Model Scoring and Evaluation** and link it to the two **Score Model** modules.

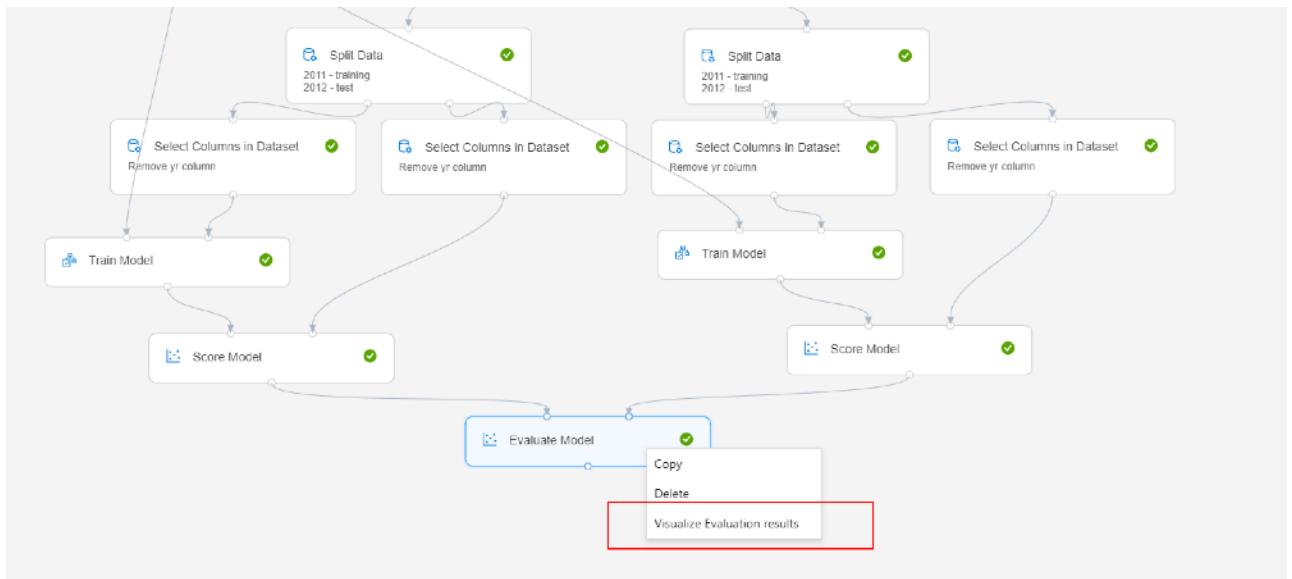


3. Select **Submit** to open the **Setup pipeline run editor**. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: BikeRentalHourly**.

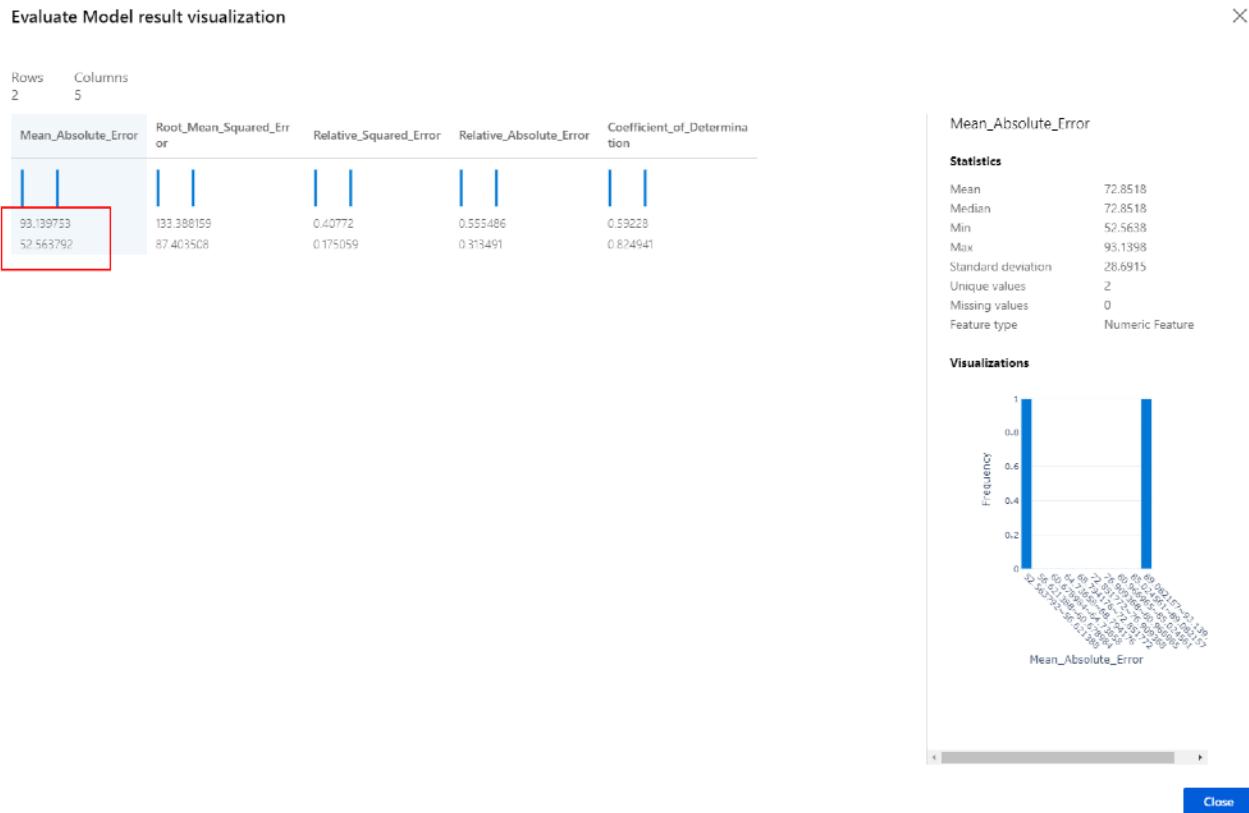


Please note that the button name in the UI is changed from **Run** to **Submit**.

4. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.
5. Once the pipeline execution completes, right click on the **Evaluate Model** module and select **Visualize Evaluation results**.



6. The **Evaluate Model result visualization** popup shows the results of the evaluation.



Notice the values for the **Mean_Absolute_Error** metric. The first value (the bigger one) corresponds to the model trained on feature set A. The second value (the smaller one) corresponds to the model trained on feature sets A + B.

It is remarkable how, using simple feature engineering to derive new features from the existing data set, a new context was created that allowed the model to better understand the dynamics of the data and hence, produce a better prediction.

Next Steps

Congratulations! You have trained and compared performance of two models using the same algorithm, but with two different datasets. 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.

Lab 5 - Train and Evaluate a Model

Overview

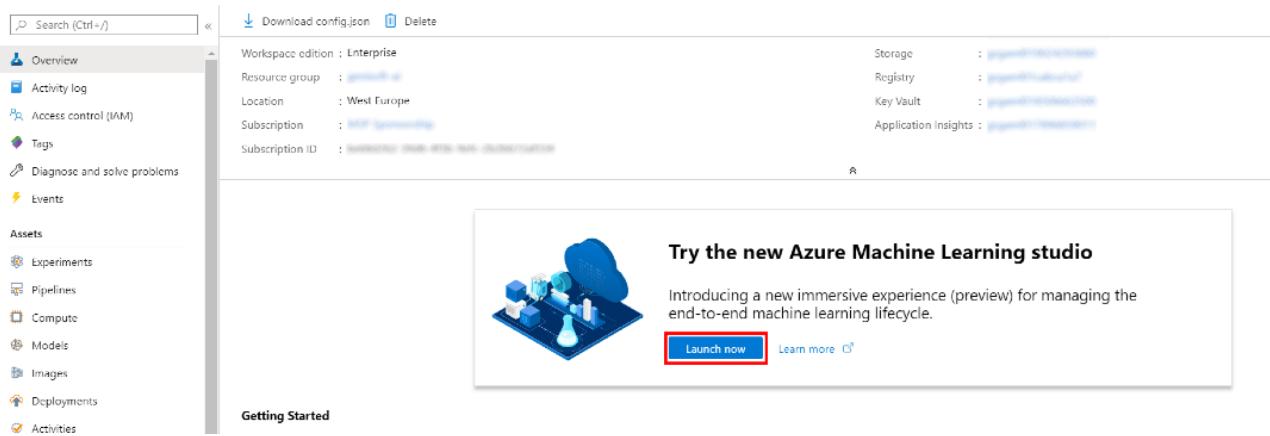
Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be using the [Flight Delays](#) data set that is enhanced with the weather data. Based on the enriched dataset, we will learn to use the Azure Machine Learning Graphical Interface to process data, build, train, score, and evaluate a classification model to predict if a particular flight will be delayed by 15 minutes or more. To train the model, we will use Azure Machine Learning Compute resource. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124

aml-quickstarts-190124

southcentralus

Get started

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**.

4. From the studio, select **Datasets, + Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the 'Datasets' page in the Azure Machine Learning Studio. The left sidebar has a 'Datasets' link highlighted with a red box. The main area shows a table with columns: Version, Created on, Modified on, Properties, and Tags. Below the table, it says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'. The 'Create dataset' dropdown menu is open, and the 'From web files' option is highlighted with a red box.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampleddata.blob.core.windows.net/data/flightdelays/flightdelays.csv>

6. Provide **flightdelays** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog box. The 'Basic info' tab is selected. The 'Web URL' field contains the URL `https://introtomlsampleddata.blob.core.windows.net/data/flightdelays/flightdelays.csv`. The 'Name' field is set to `flightdelays`. The 'Dataset type' dropdown is set to `Tabular`. The 'Next' button is highlighted with a red box.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Review the dataset and then select **Next**

Task 3: Select Columns

- Select columns from the dataset to include as part of your training data. Leave the default selections and select **Next**

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' blade on the left and the 'Create dataset from web files' dialog on the right. The dialog has four tabs: 'Basic info' (selected), 'Settings and preview', 'Schema', and 'Confirm details'. Under 'Basic info', the dataset name is 'flightdelays', version is '1', and type is 'Tabular'. Under 'File settings', the file format is 'Delimited' with 'Comma' delimiter and 'UTF-8' encoding. A checkbox for 'Profile this dataset after creation' is present. At the bottom are 'Back', 'Create' (highlighted with a red box), and 'Cancel' buttons.

Exercise 2: Create New Training Pipeline

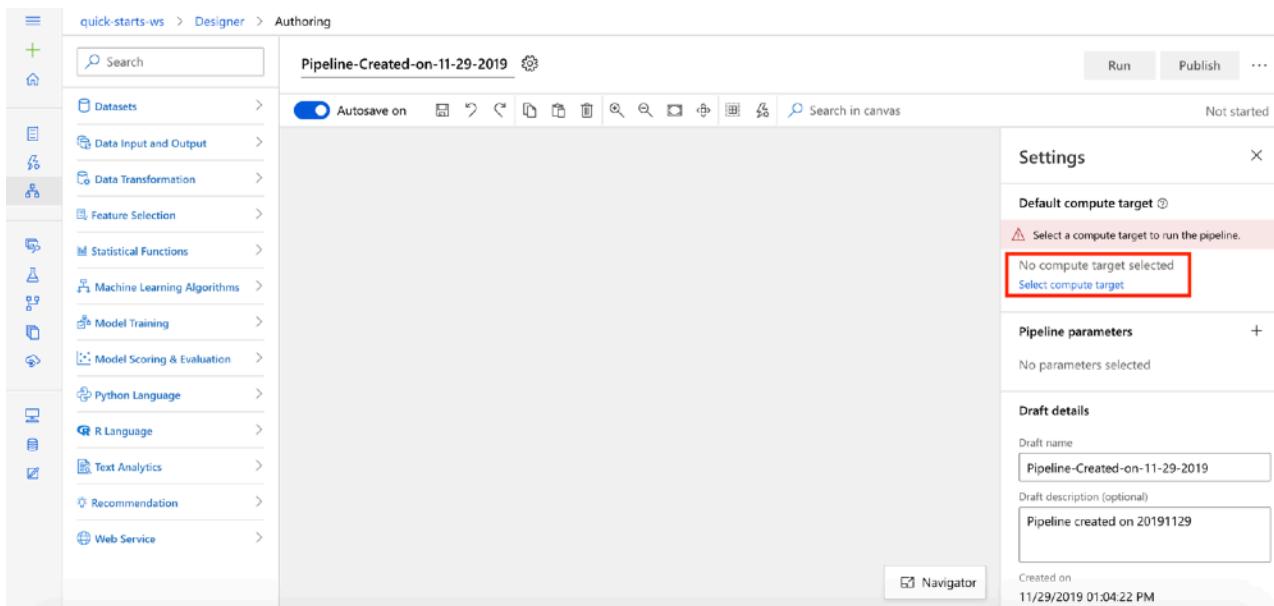
Task 1: Open Pipeline Authoring Editor

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

The screenshot shows the 'Designer' blade in the Azure Data Studio. On the left, there's a sidebar with icons. A red box highlights the '+' icon under 'Easy-to-use prebuilt modules'. Below this, there are four sample pipeline cards: 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', and 'Sample 4: Binary Classification with custom Python script - ...'. The main area shows the 'Pipelines' section with a single pipeline draft named 'Untitled Pipeline'. There are buttons for 'Refresh' and 'Delete', and a search bar at the bottom.

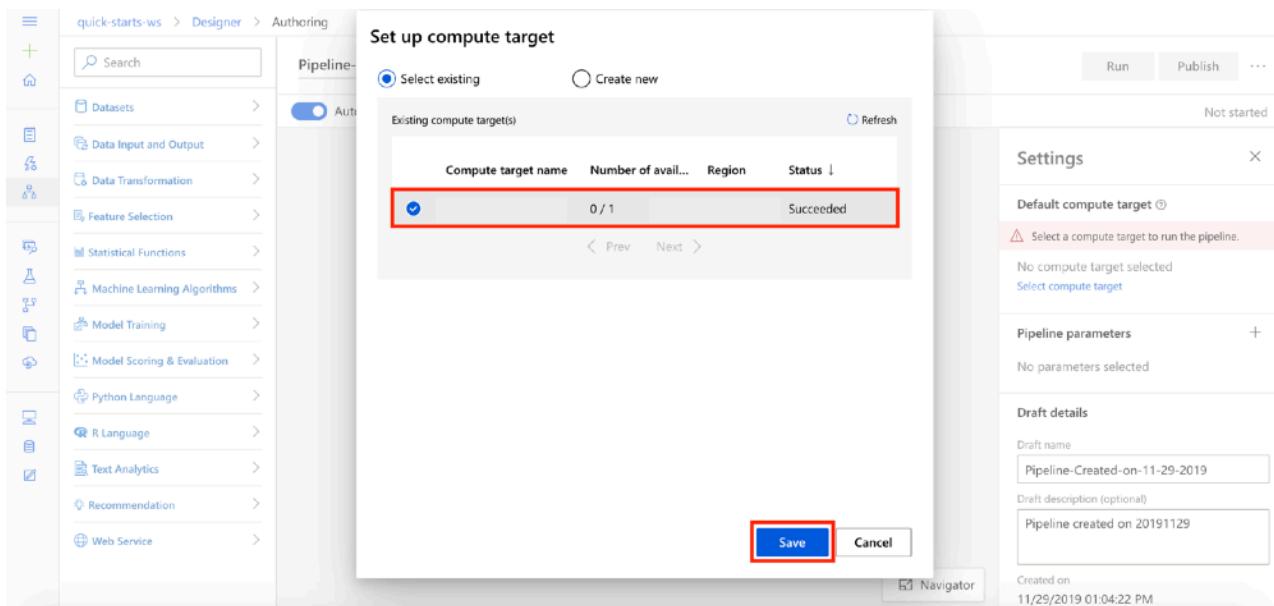
Task 2: Setup Compute Target

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



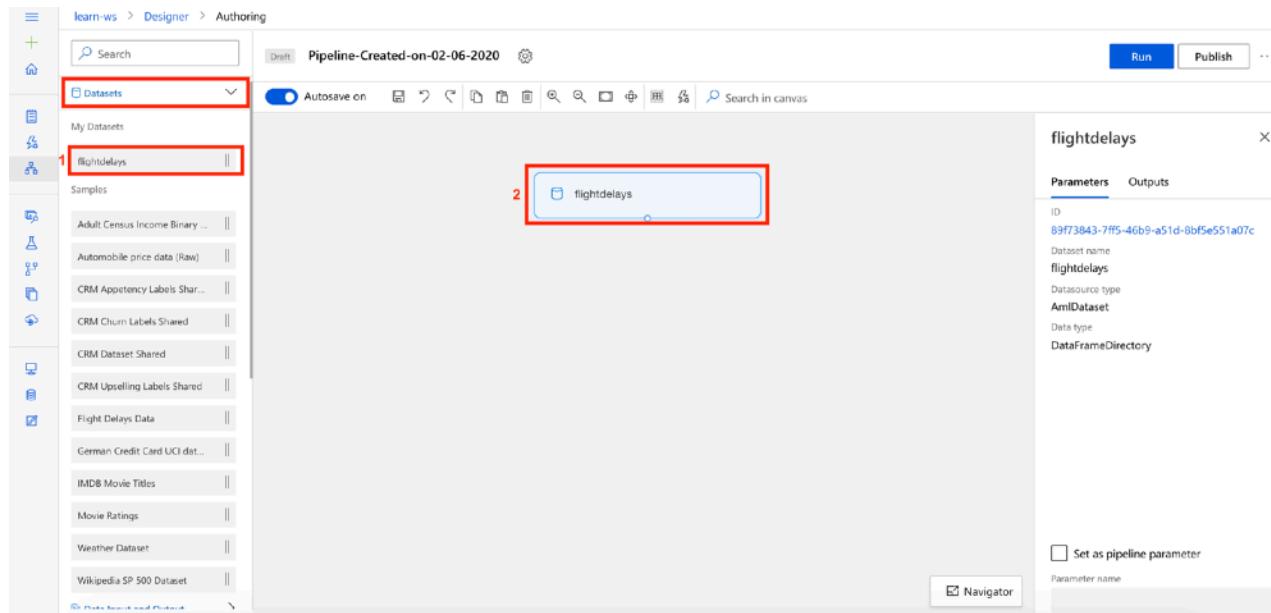
2. In the **Set up compute target** editor, select the available compute, and then select **Save**.

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.



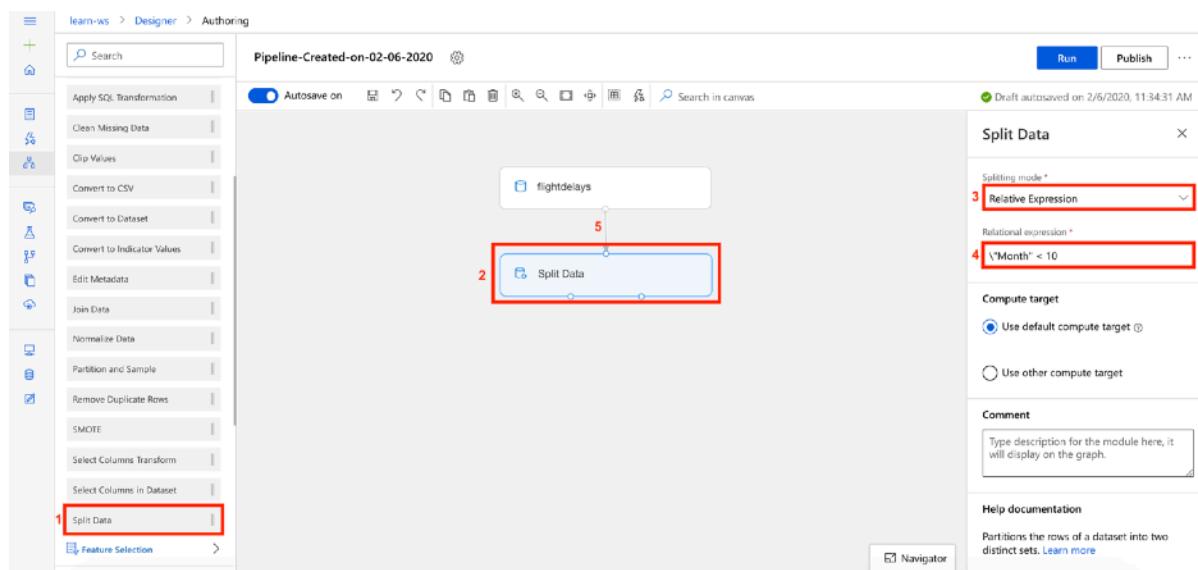
Task 3: Add Dataset

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



Task 4: Split Dataset

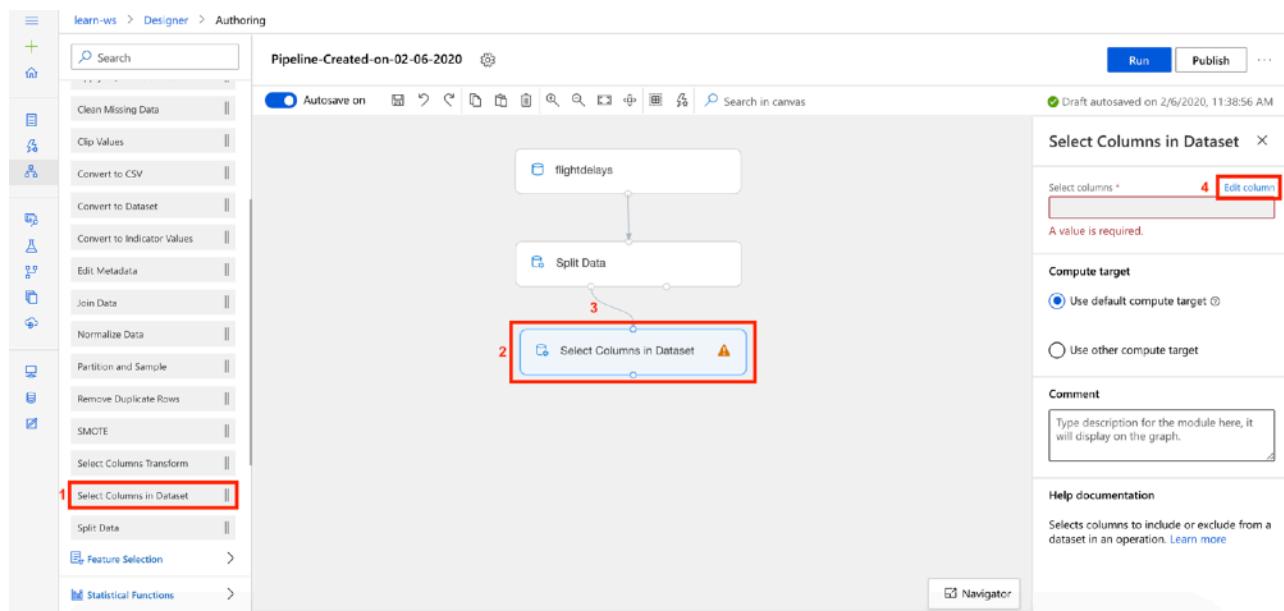
1. We will split the dataset such that months prior to October will be used for model training and months October to December will be used for model testing.
2. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Split Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Splitting mode: **Relative Expression**
 4. Relational expression: **\"Month" < 10**
 5. Connect the **Dataset** to the **Split Data** module



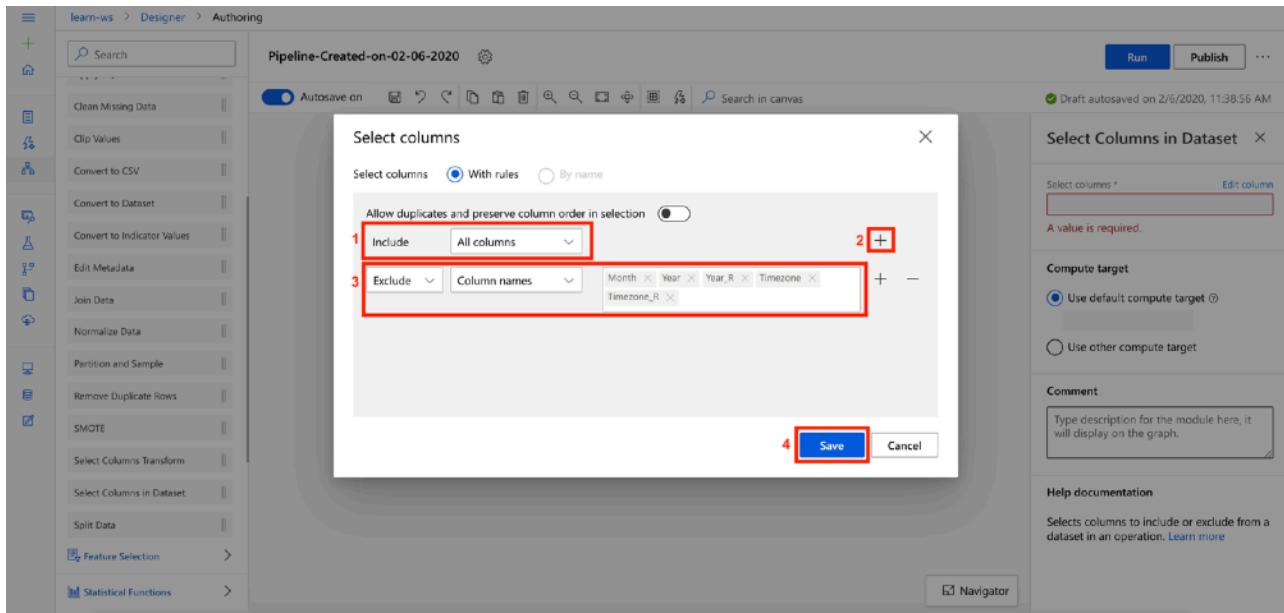
Note that you can submit the pipeline at any point to peek at the outputs and activities. Running pipeline also generates metadata that is available for downstream activities such selecting column names from a list in selection dialogs. Please refer ahead to Exercise 3, Task 1, Step 2 on details of submitting the pipeline. It can take up to 5-10 minutes to run the pipeline.

Task 5: Select Columns in Dataset

- Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 - Select the **Select Columns in Dataset** prebuilt module
 - Drag and drop the selected module on to the canvas
 - Connect the first output of the **Split Data** module to the **Select Columns in Dataset** module
 - Select **Edit column** link to open the Select columns` editor

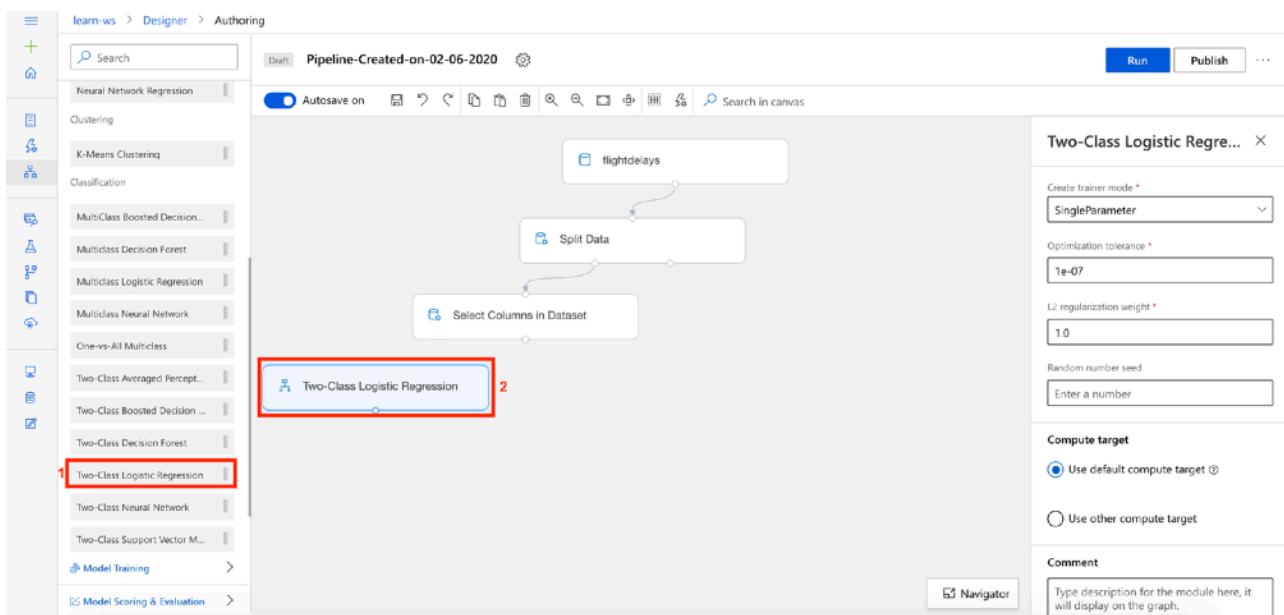


- In the **Select columns** editor, follow the steps outlined below:
 - Include: **All columns**
 - Select **+**
 - Exclude: **Column names**, provide the following column names to exclude: **Month, Year, Year_R, Timezone, Timezone_R**
 - Select **Save**



Task 6: Initialize Classification Model

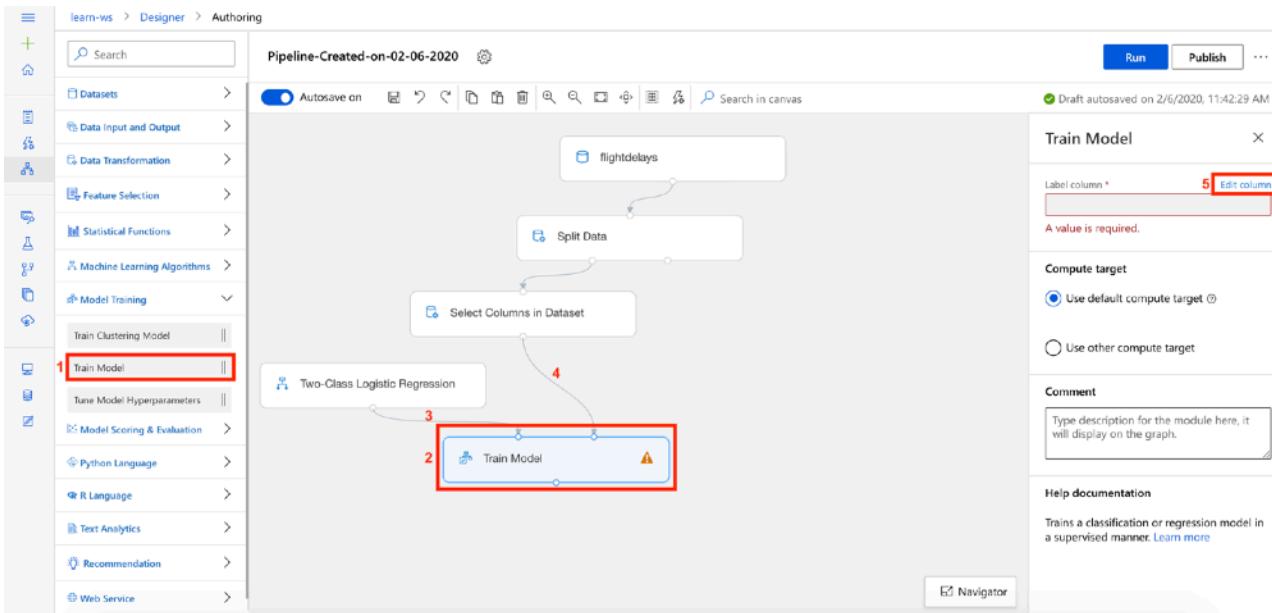
1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Two-Class Logistic Regression** prebuilt module
 2. Drag and drop the selected module on to the canvas



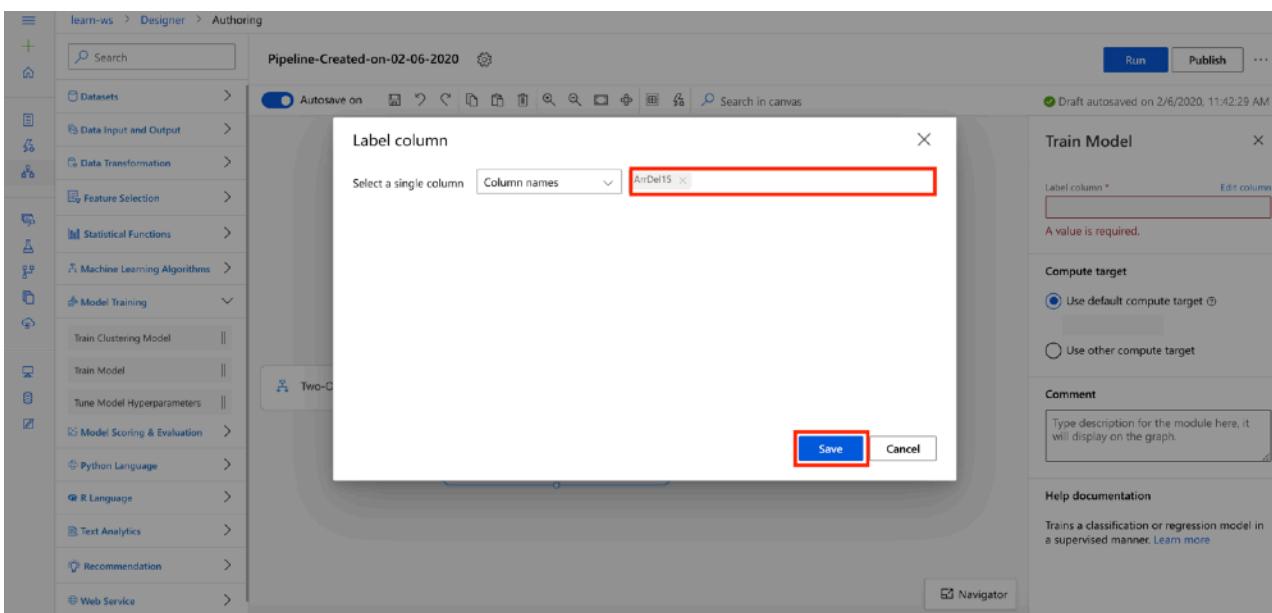
Task 7: Setup Train Model Module

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Model** prebuilt module
 2. Drag and drop the selected module on to the canvas

3. Connect the **Two-Class Logistic Regression** module to the first input of the **Train Model** module
4. Connect the **Select Columns in Dataset** module to the second input of the **Train Model** module
5. Select the **Edit column** link to open the **Label column** editor



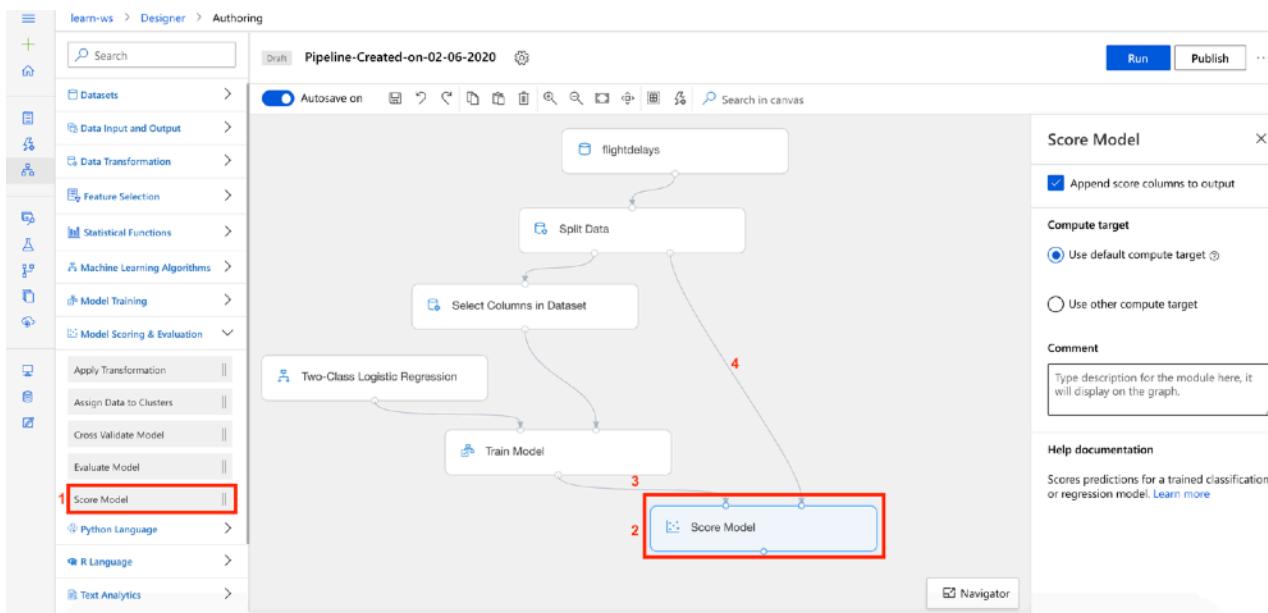
2. The **Label column** editor allows you to specify your **Label or Target column**. Type in the label column name **ArrDel15** and then select **Save**.



Task 8: Setup Score Model Module

1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas

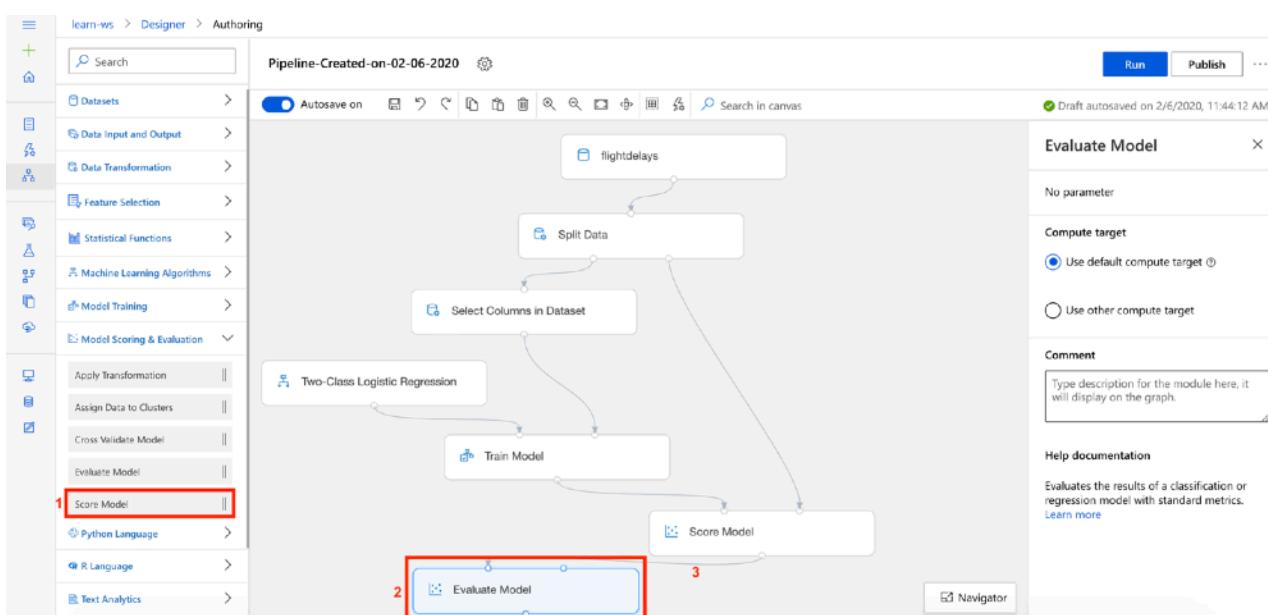
3. Connect the **Train Model** module to the first input of the **Score Model** module
4. Connect the second output of the **Split Data** module to the second input of the **Score Model** module



Note that **Split Data** module will feed data for both model training and model scoring.

Task 9: Setup Evaluate Model Module

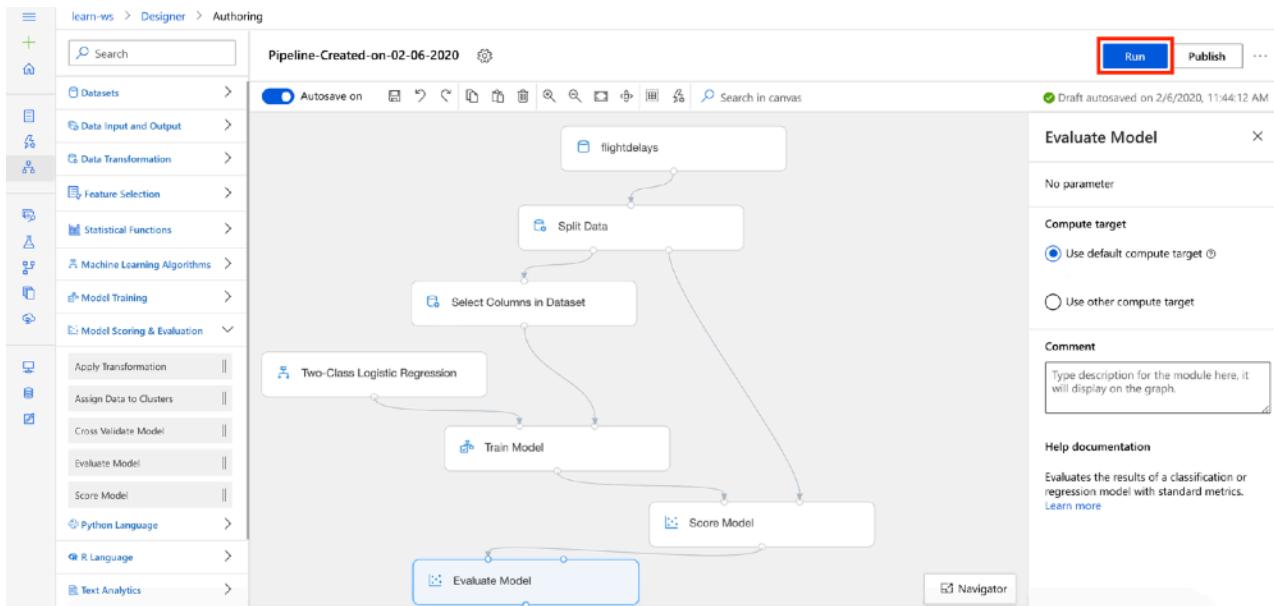
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Score Model** module to the first input of the **Evaluate Model** module



Exercise 3: Submit Training Pipeline

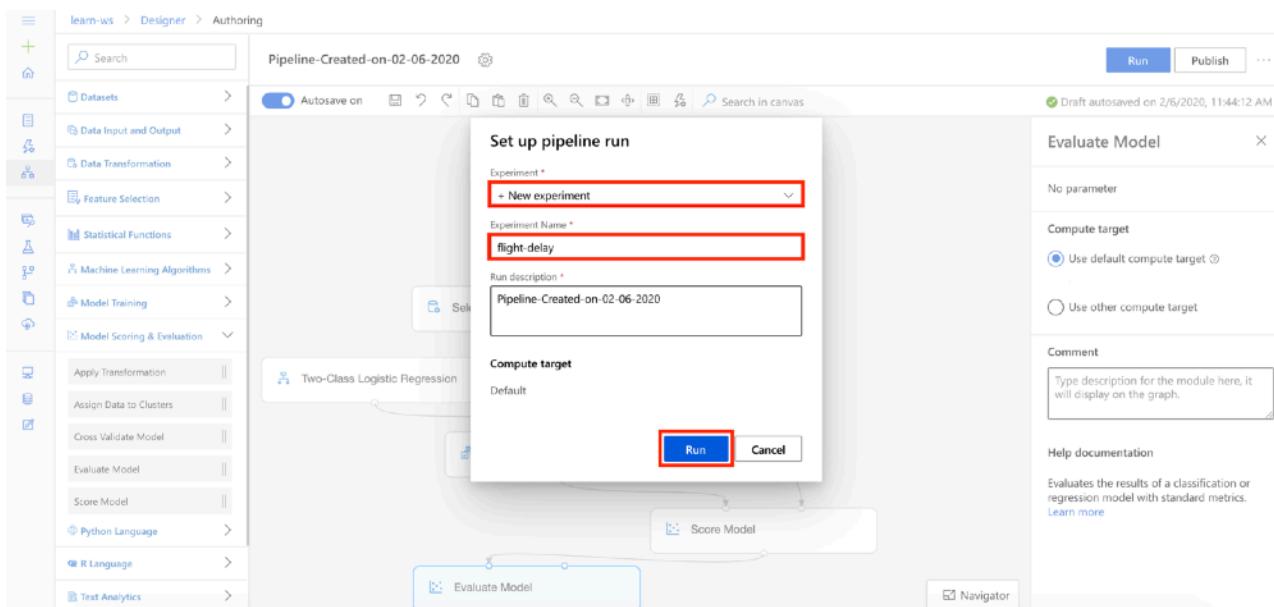
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



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

2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: flight-delay**, and then select **Submit**.

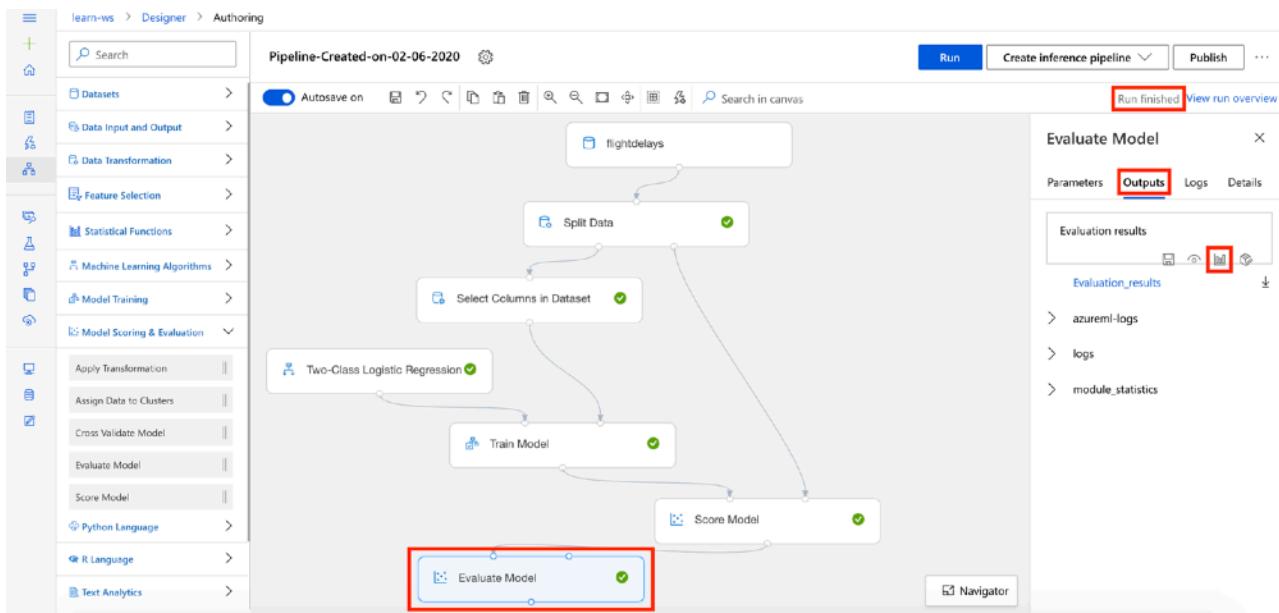


3. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the evaluation metrics for the classification algorithm used in this lab by selecting [Metrics for classification models](#).

Exercise 4: Visualize the Evaluation Results

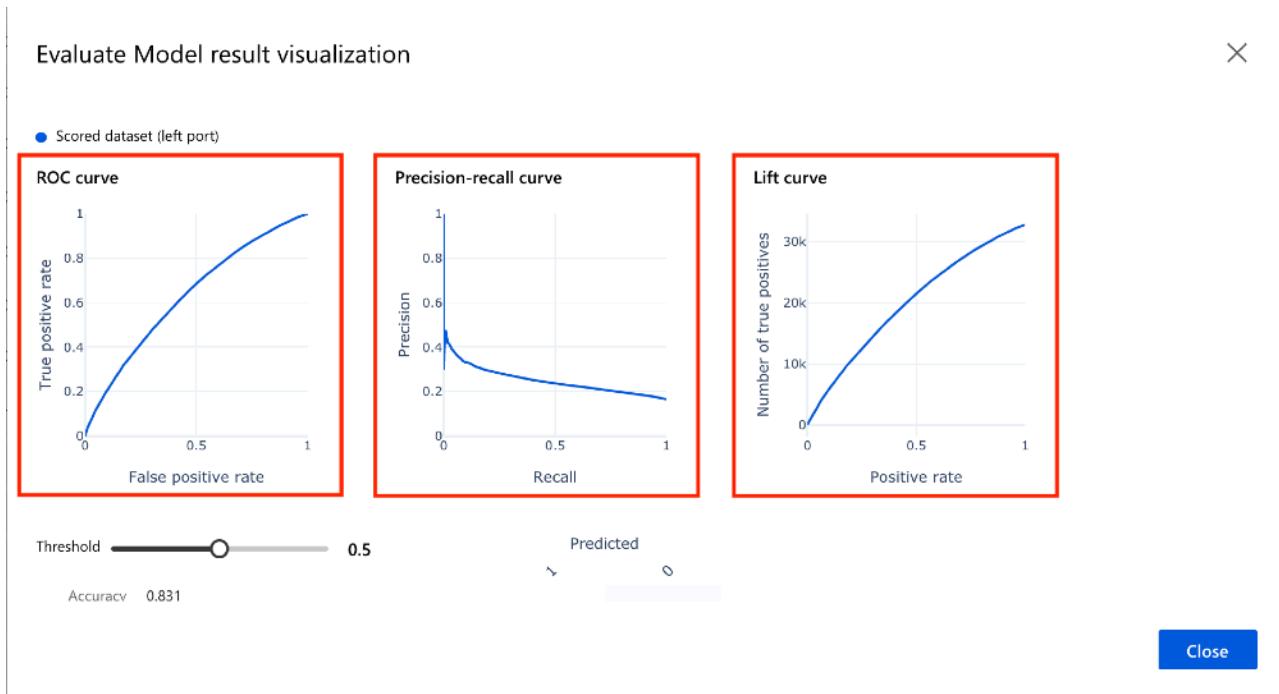
Task 1: Open the Result Visualization Dialog

1. Select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog.



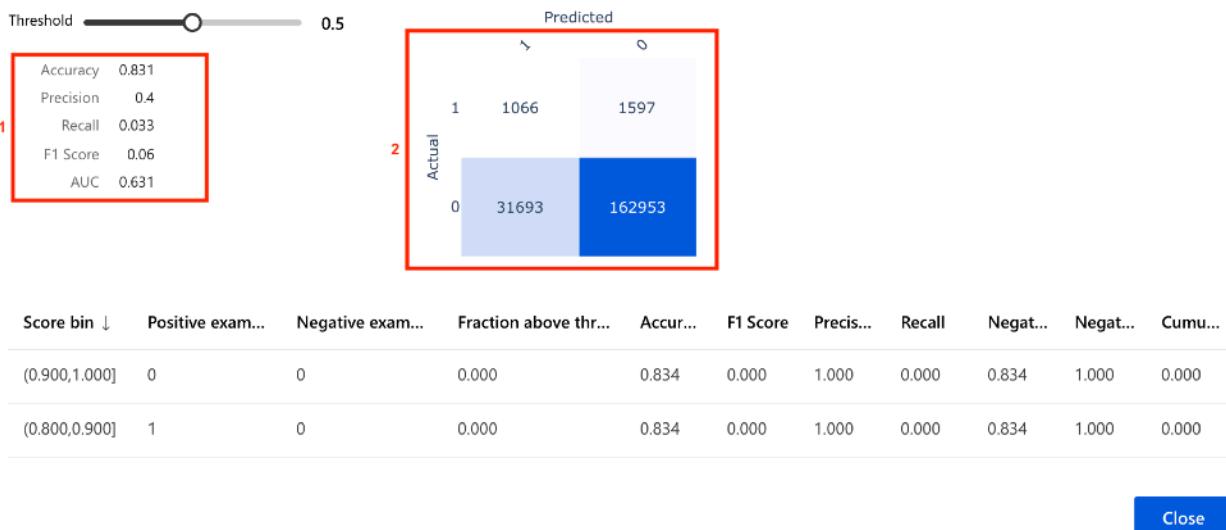
Task 2: Evaluate Model Performance

1. Evaluate the model performance by reviewing the various evaluation curves, such as **ROC curve**, **Precision-recall curve**, and **Lift curve**.



2. Scroll down to review the following:
 1. Review the key metrics for classifiers: **Accuracy, Precision, Recall, F1 Score, and AUC**
 2. Review the binary classifier's **Confusion Matrix**

Evaluate Model result visualization



Next Steps

Congratulations! You have trained and evaluated your first classification machine learning model. 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.

Lab 6 - Train a Two-Class Boosted Decision Tree

Overview

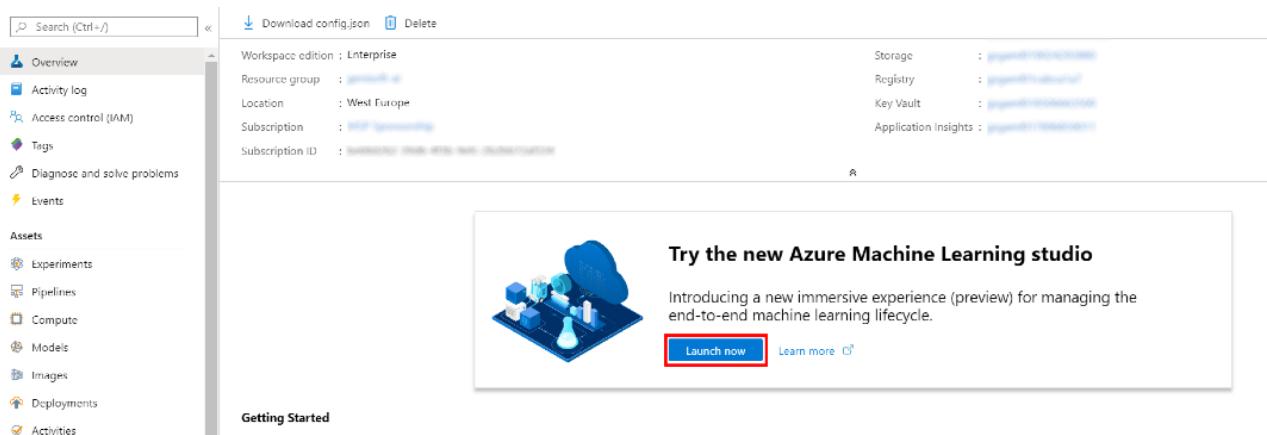
[Azure Machine Learning designer](#) (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be using the [Flight Delays](#) data set that is enhanced with the weather data. Based on the enriched dataset, we will learn to use the Azure Machine Learning Graphical Interface to process data, build, train, score, and evaluate a classification model to predict if a particular flight will be delayed by 15 minutes or more. The classification algorithm used in this lab will be the ensemble algorithm: **Two-Class Boosted Decision Tree**. To train the model, we will use Azure Machine Learning Compute resource. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124
aml-quickstarts-190124

southcentralus

Get started

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**.

4. From the studio, select **Datasets**, + **Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the Azure Machine Learning Studio interface. On the left, there's a sidebar with various options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets' (which is highlighted with a red box), 'Experiments', 'Pipelines', 'Models', and 'Endpoints'. The main area is titled 'Datasets' and shows 'Registered datasets' and 'Dataset monitors (Preview)'. There's a 'Create dataset' dropdown with options: 'From local files', 'From datastore', 'From web files' (which is also highlighted with a red box), and 'From Open Datasets'. Below these options, it says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampled.blob.core.windows.net/data/flightdelays/flightdelays.csv>

6. Provide **flightdelays** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there are tabs: 'Basic info' (selected), 'Settings and preview', 'Schema', and 'Confirm details'. The 'Basic info' tab has fields for 'Web URL' (containing the URL 'https://introtomlsampled.blob.core.windows.net/data/flightdelays/flightdelays.csv'), 'Name' (containing 'flightdelays'), 'Dataset type' (set to 'Tabular'), and a 'Description' field. At the bottom, there are 'Back', 'Next' (which is highlighted with a red box), and 'Cancel' buttons.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Review the dataset and then select **Next**

Task 3: Select Columns

- Select columns from the dataset to include as part of your training data. Leave the default selections and select **Next**

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' blade on the left and the 'Create dataset from web files' dialog on the right. The dialog has four tabs: 'Basic info' (selected), 'Settings and preview', 'Schema', and 'Confirm details'. Under 'Basic info', fields include Name (flightdelays), Version (1), Dataset type (Tabular), and Web URL (https://introtomlsampledata.blob.core.windows.net/data/flightdelays/flightdelays.csv). Under 'File settings', options are File format (Delimited), Delimiter (Comma), Encoding (UTF-8), Column headers (All files have same headers), and Skip rows (None). At the bottom, there's a 'Profile this dataset after creation' checkbox, a 'Back' button, a red-bordered 'Create' button, and a 'Cancel' button.

Exercise 2: Create New Training Pipeline

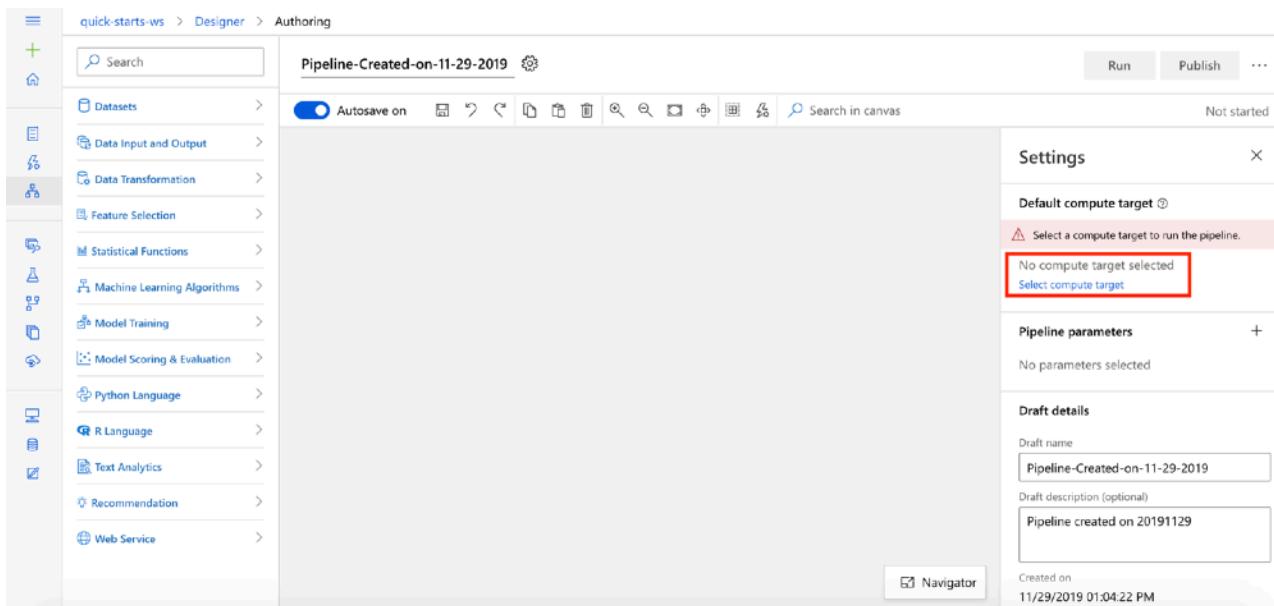
Task 1: Open Pipeline Authoring Editor

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

The screenshot shows the 'Designer' blade in the Azure Data Studio. On the left, there's a sidebar with icons for Datasets, Pipelines, and more. The main area has a 'New pipeline' section with a large red-bordered '+' button. Below it are four sample pipeline cards: 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', and 'Sample 4: Binary Classification with custom Python script - ...'. In the 'Pipelines' section, there's a table with one row for 'Untitled Pipeline'. At the bottom, there's a search bar and a 'Search to filter items...' placeholder.

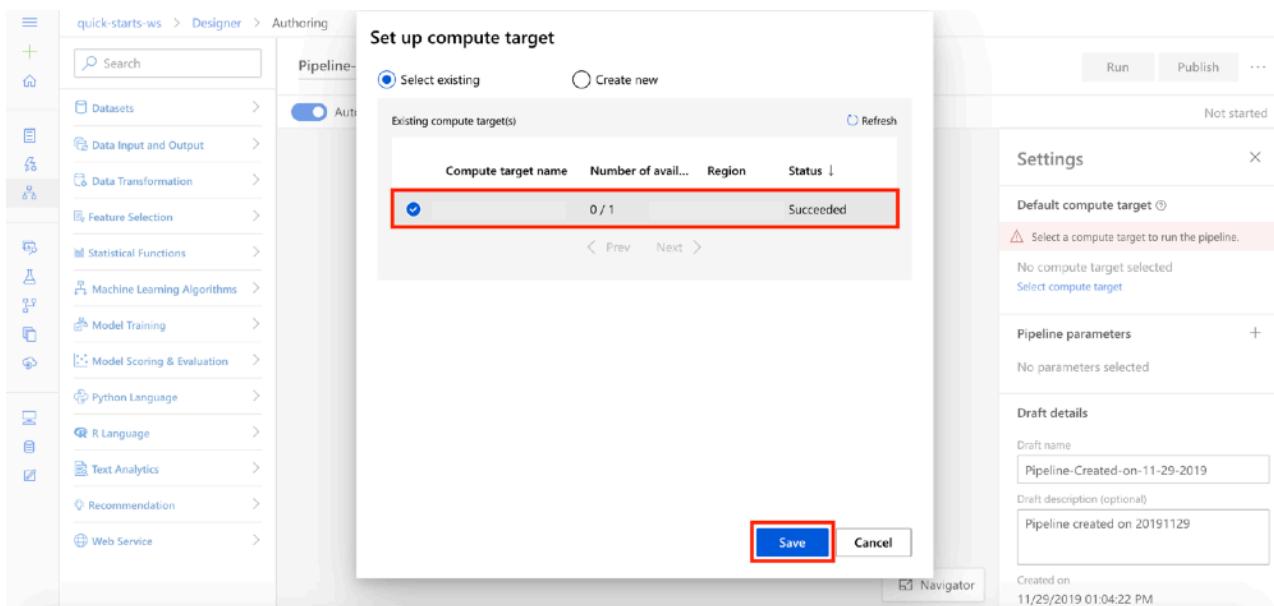
Task 2: Setup Compute Target

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



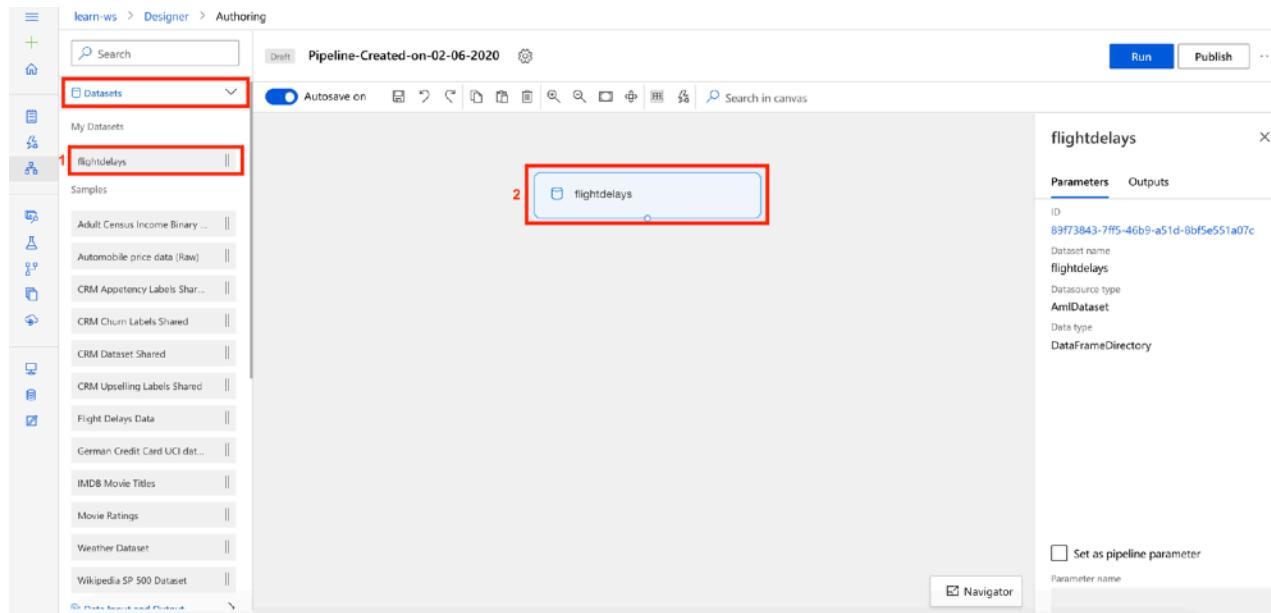
2. In the **Set up compute target** editor, select the available compute, and then select **Save**.

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.



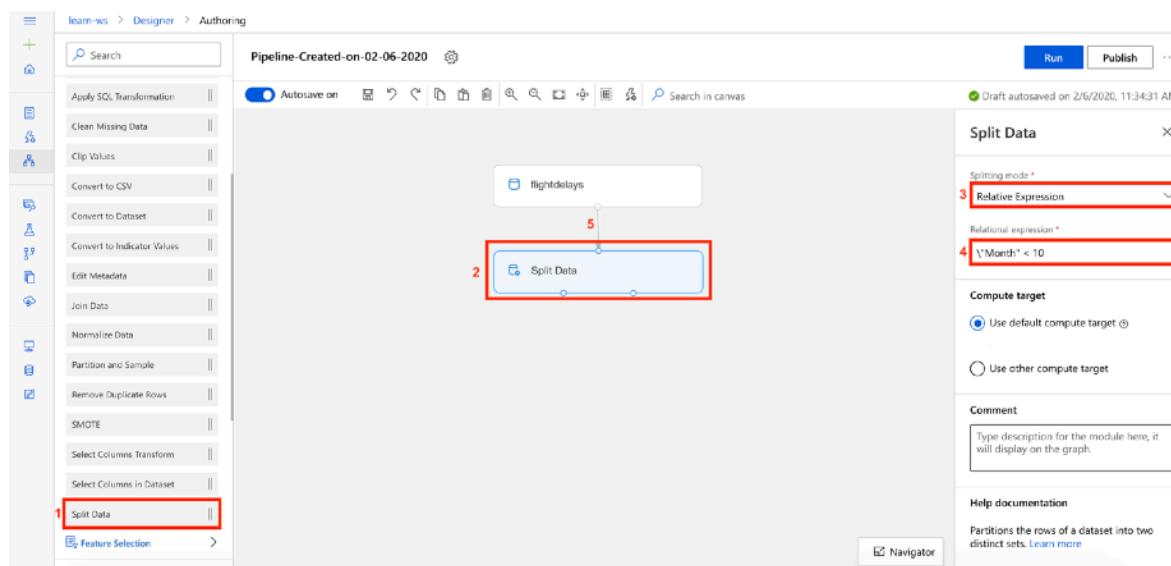
Task 3: Add Dataset

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



Task 4: Split Dataset

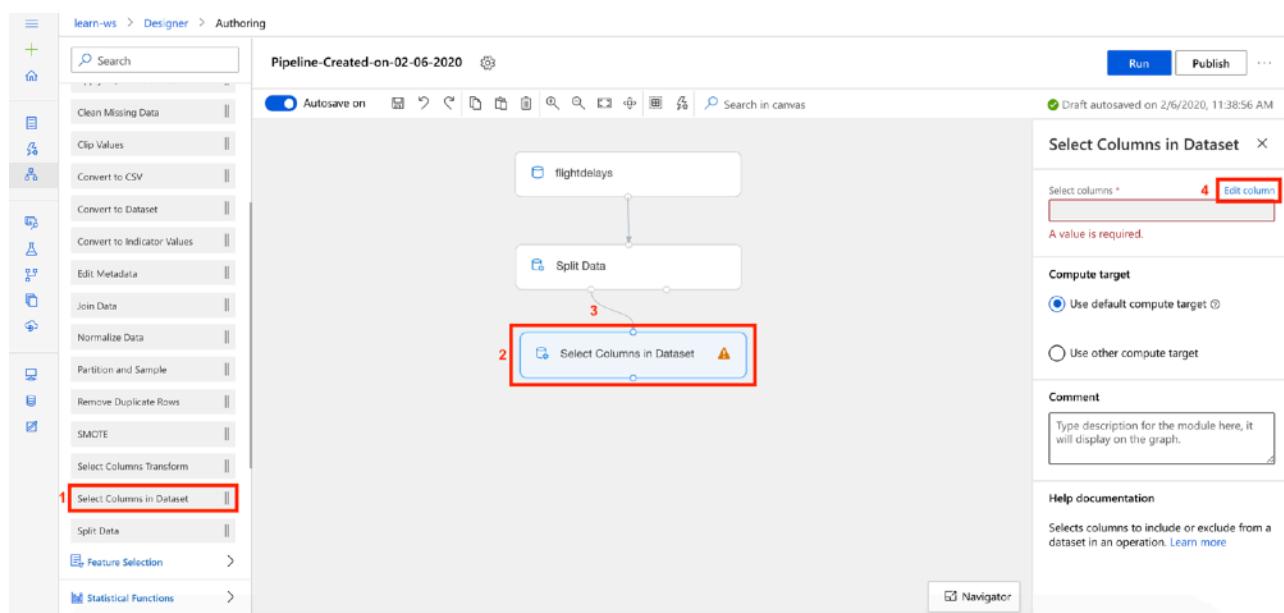
1. We will split the dataset such that months prior to October will be used for model training and months October to December will be used for model testing.
2. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Split Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Splitting mode: **Relative Expression**
 4. Relational expression: **\"Month" < 10**
 5. Connect the **Dataset** to the **Split Data** module



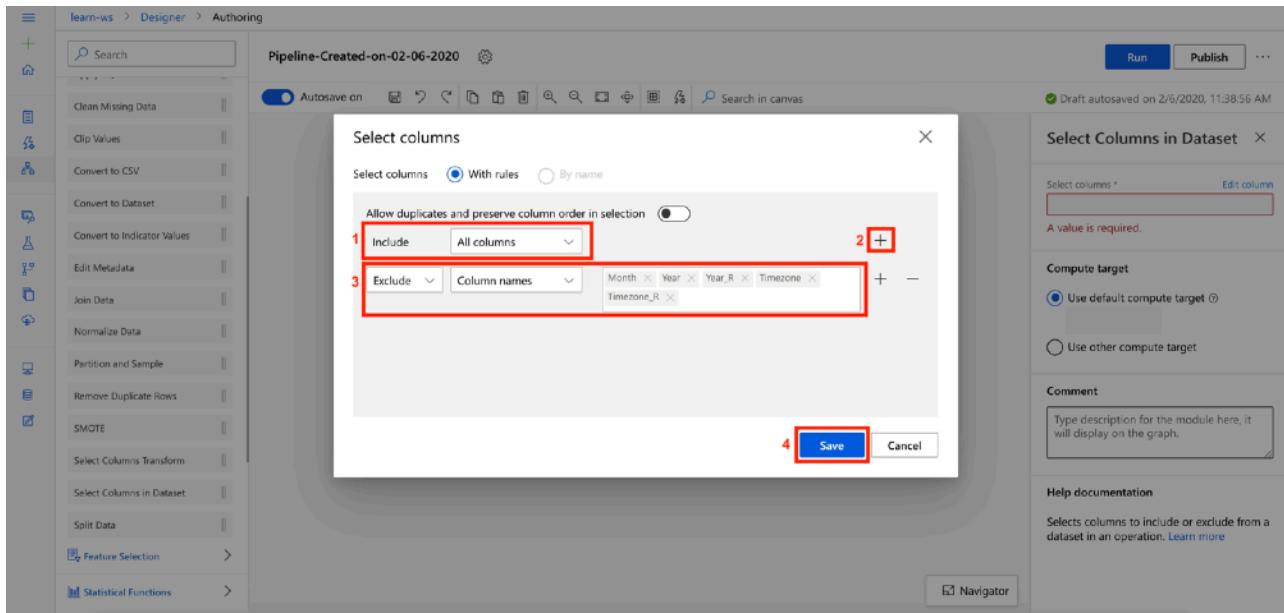
Note that you can submit the pipeline at any point to peek at the outputs and activities. Running pipeline also generates metadata that is available for downstream activities such selecting column names from a list in selection dialogs.

Task 5: Select Columns in Dataset

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 first output of the **Split Data** module to the **Select Columns in Dataset** module
 4. Select **Edit column** link to open the Select columns` editor

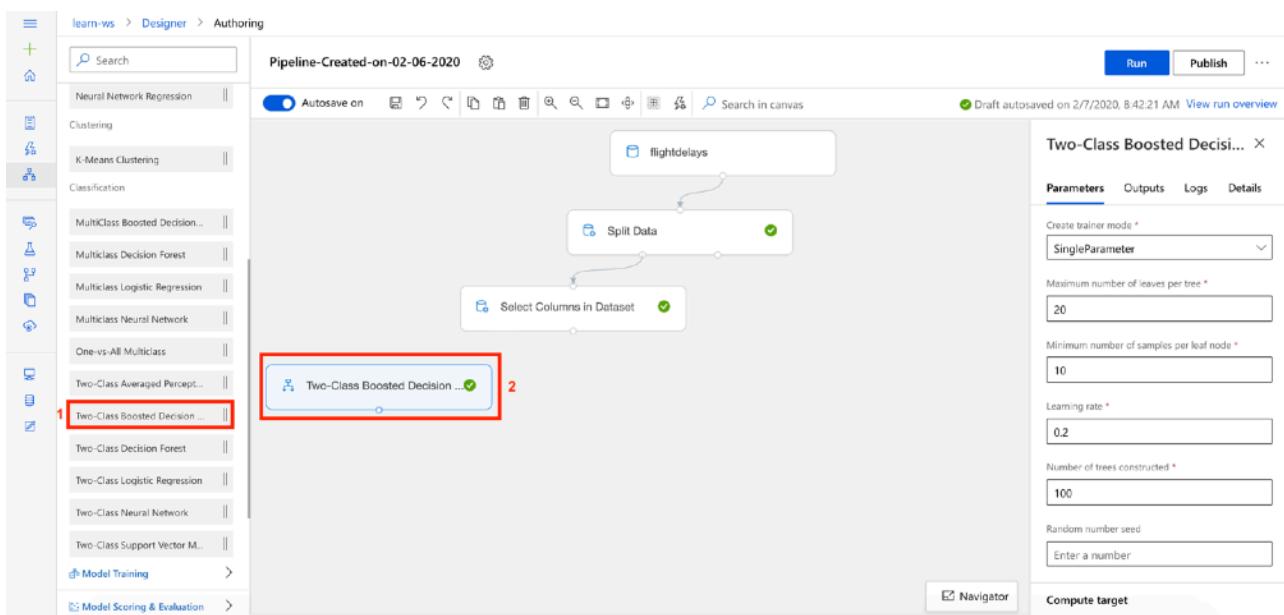


2. In the **Select columns** editor, follow the steps outlined below:
 1. Include: **All columns**
 2. Select **+**
 3. Exclude: **Column names**, provide the following column names to exclude: **Month, Year, Year_R, Timezone, Timezone_R**
 4. Select **Save**



Task 6: Initialize Classification Model

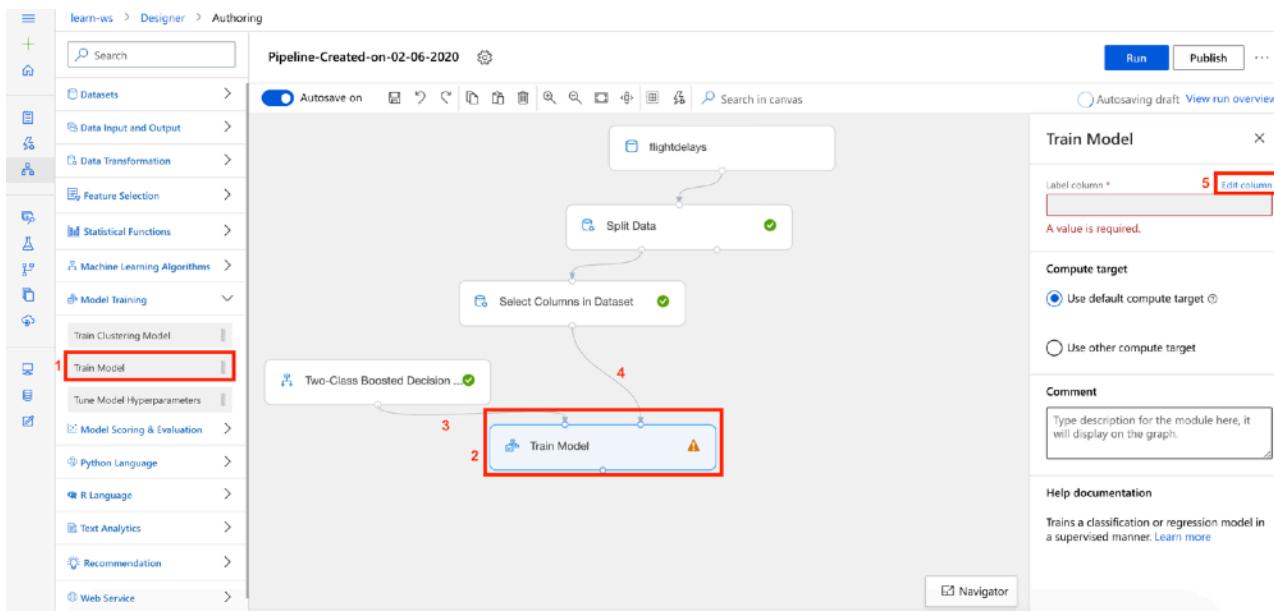
1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Two-Class Boosted Decision Tree** prebuilt module
 2. Drag and drop the selected module on to the canvas



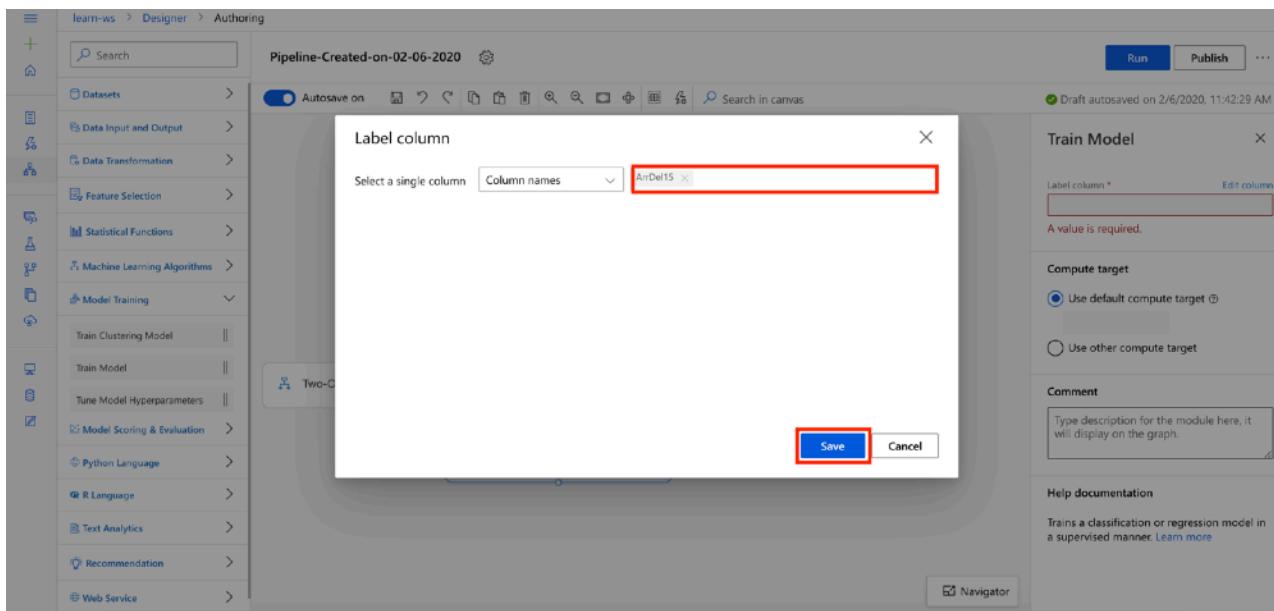
Task 7: Setup Train Model Module

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Model** prebuilt module
 2. Drag and drop the selected module on to the canvas

3. Connect the **Two-Class Boosted Decision Tree** module to the first input of the **Train Model** module
4. Connect the **Select Columns in Dataset** module to the second input of the **Train Model** module
5. Select the **Edit column** link to open the **Label column** editor



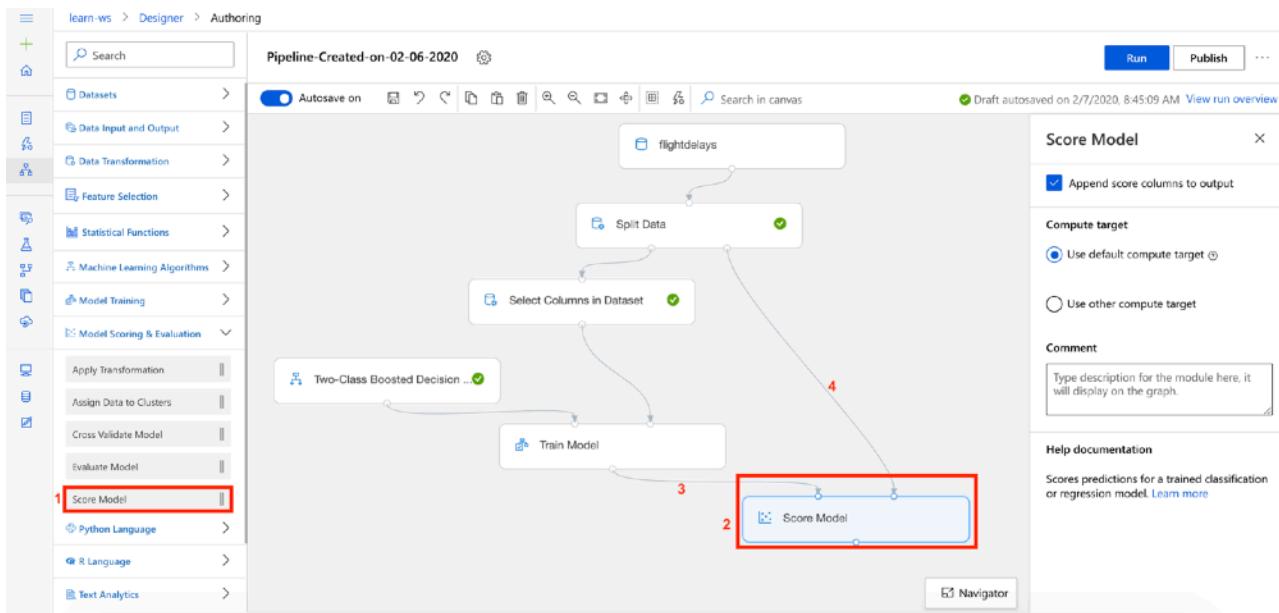
2. The **Label column** editor allows you to specify your **Label or Target column**. Type in the label column name **ArrDel15** and then select **Save**.



Task 8: Setup Score Model Module

1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas

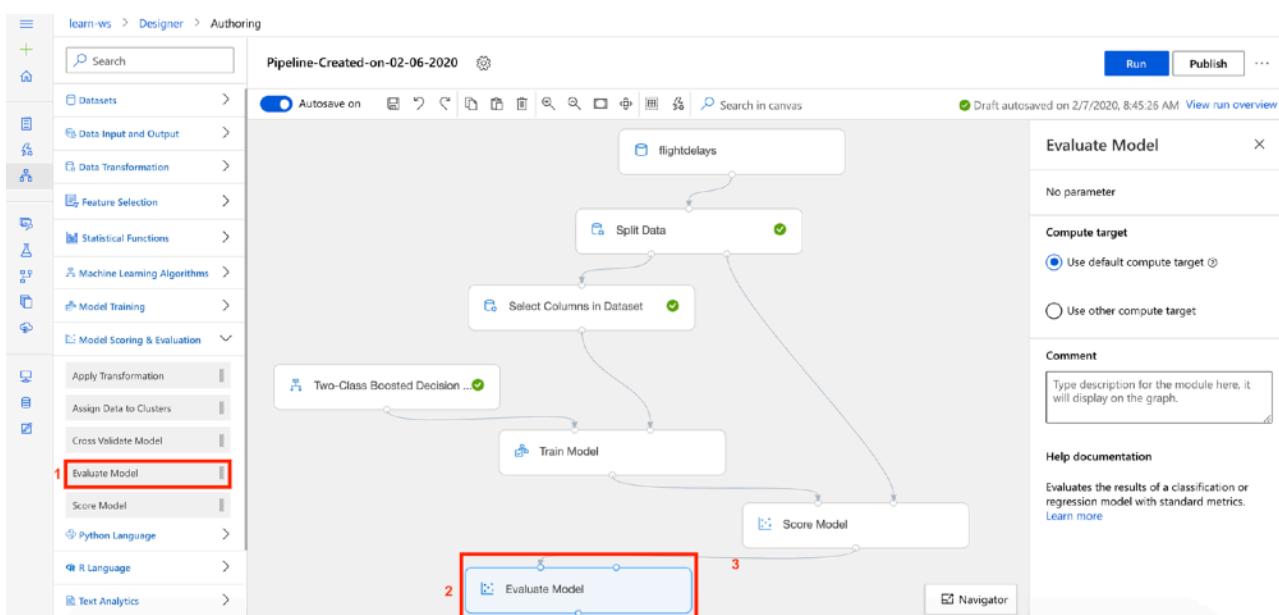
3. Connect the **Train Model** module to the first input of the **Score Model** module
4. Connect the second output of the **Split Data** module to the second input of the **Score Model** module



Note that **Split Data** module will feed data for both model training and model scoring.

Task 9: Setup Evaluate Model Module

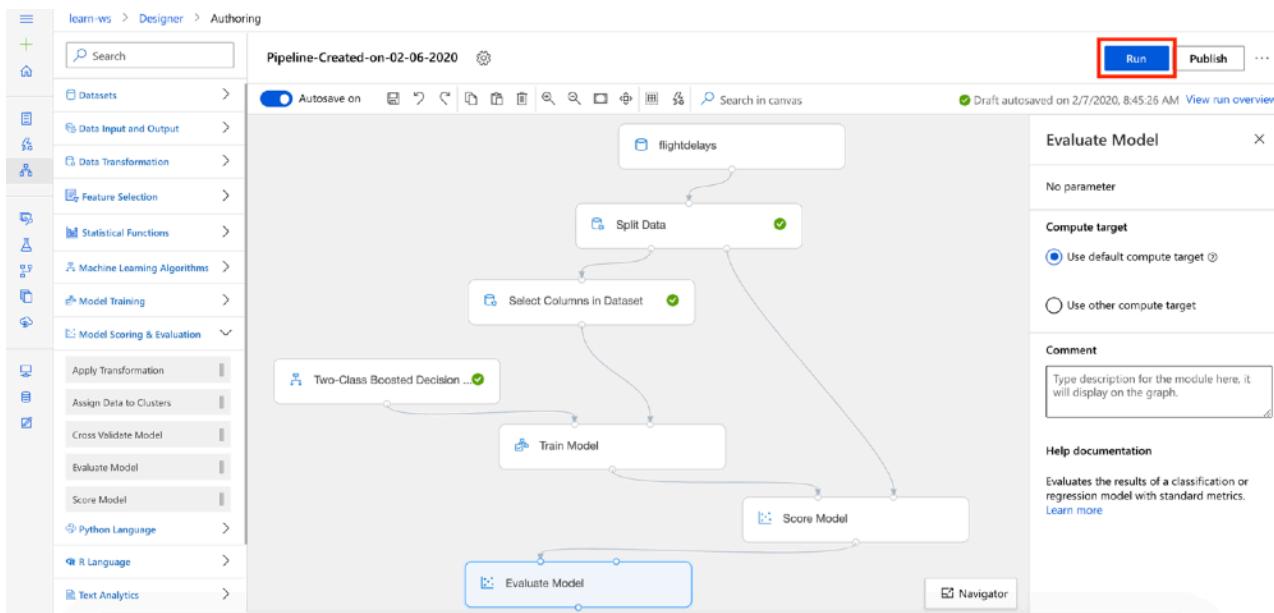
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Score Model** module to the first input of the **Evaluate Model** module



Exercise 3: Submit Training Pipeline

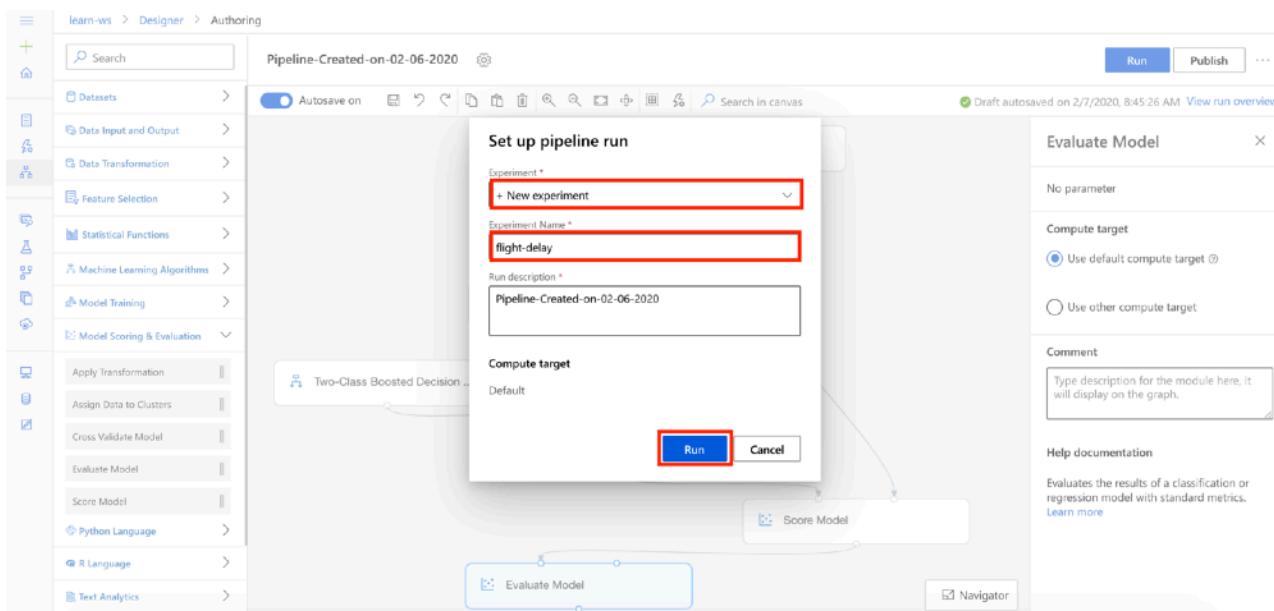
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



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

2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: flight-delay**, and then select **Submit**.

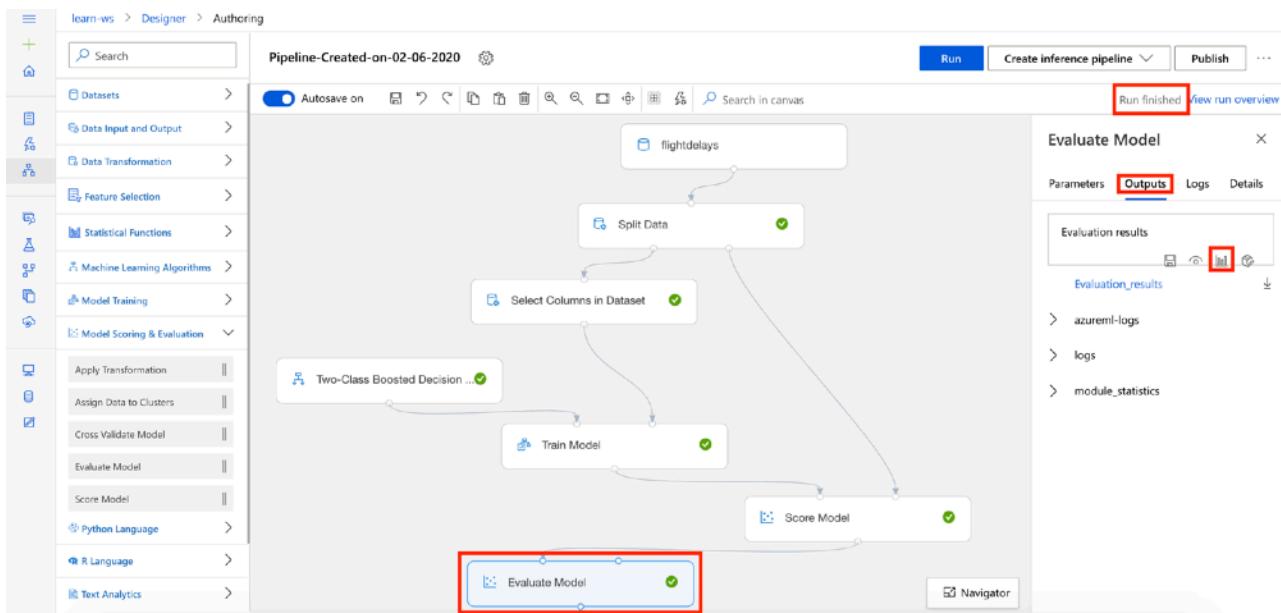


3. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the classification algorithm used in this lab by selecting **Two-Class Boosted Decision Tree**.

Exercise 4: Visualize the Evaluation Results

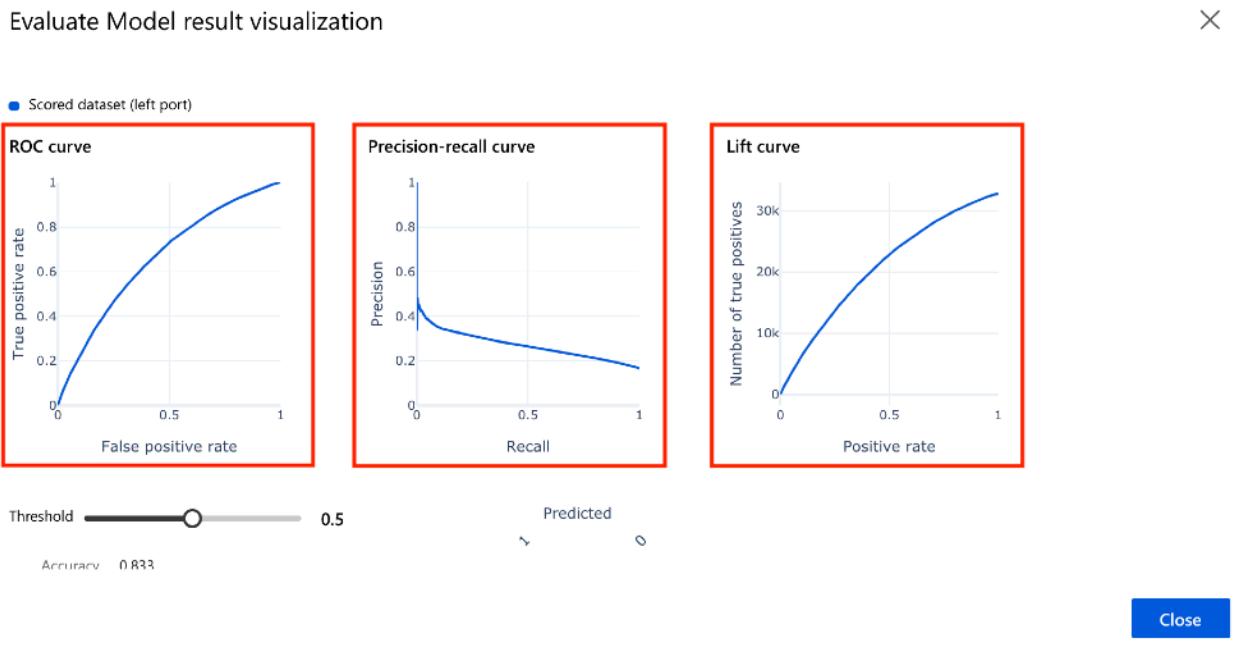
Task 1: Open the Result Visualization Dialog

1. Select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog.



Task 2: Evaluate Model Performance

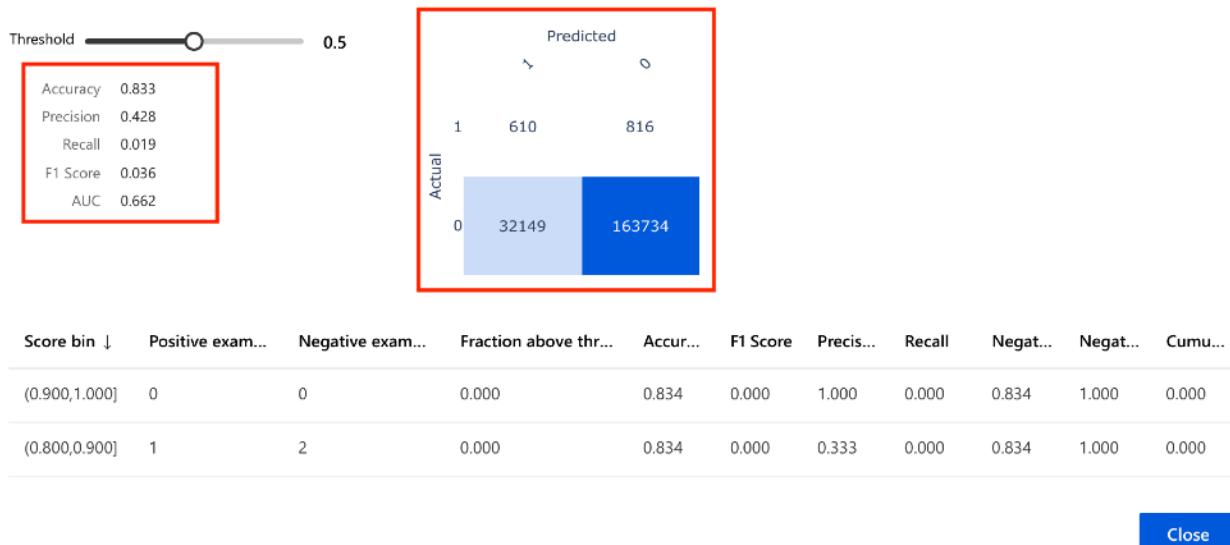
1. Evaluate the model performance by reviewing the various evaluation curves, such as **ROC curve**, **Precision-recall curve**, and **Lift curve**.



2. Scroll down to review the following:
 1. Review the key metrics for classifiers: **Accuracy, Precision, Recall, F1 Score, and AUC**
 2. Review the binary classifier's **Confusion Matrix**

Evaluate Model result visualization

X



Next Steps

Congratulations! You have trained and evaluated your first ensemble machine learning model. 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.

Lab 7 - Train a Simple Classifier with Automated ML

Overview

Automated machine learning picks an algorithm and hyperparameters for you and generates a model ready for deployment. There are several options that you can use to configure automated machine learning experiments.

Configuration options available in automated machine learning:

- Select your experiment type: Classification, Regression or Time Series Forecasting
- Data source, formats, and fetch data
- Choose your compute target
- Automated machine learning experiment settings
- Run an automated machine learning experiment
- Explore model metrics
- Register and deploy model

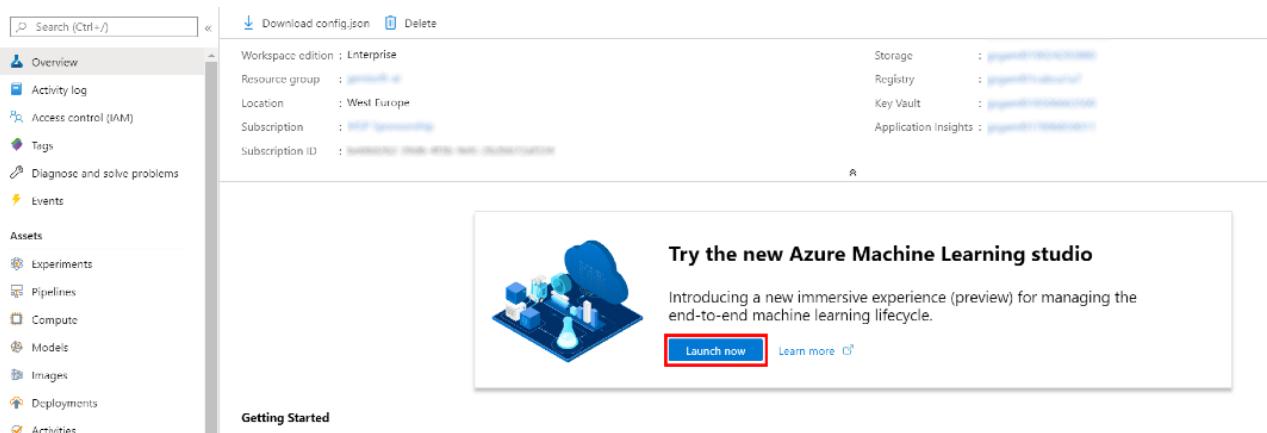
You can create and run automated machine learning experiments in code using the [Azure ML Python SDK](#) or if you prefer a no code experience, you can also create your automated machine learning experiments in [Azure Machine Learning Studio](#).

In this lab, you learn how to create, run, and explore automated machine learning experiments in the [Azure Machine Learning Studio](#) without a single line of code. As part of this lab, we will be using the [Flight Delays](#) data set that is enhanced with the weather data. Based on the enriched dataset, we will use automated machine learning to find the best performing classification model to predict if a particular flight will be delayed by 15 minutes or more.

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

The screenshot shows the "Welcome to the studio!" setup screen. It has three dropdown menus:

- Switch directory:** The dropdown is set to "Udacity".
- Subscription:** The dropdown is set to "Azure Sponsorship - Udacity -04".
- Machine learning workspace:** The dropdown is set to "quick-starts-ws-190124". Below it, other options are listed: "quick-starts-ws-190124" and "aml-quickstarts-190124".

A blue "Get started" button is located at the bottom of the dropdown menu.

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**.

4. From the studio, select **Datasets**, + **Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the 'Datasets' page in the Azure Machine Learning Studio. On the left, there's a sidebar with various options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets' (which is selected and highlighted with a red box), 'Experiments', 'Pipelines', 'Models', and 'Endpoints'. The main area is titled 'Datasets' and has tabs for 'Registered datasets' and 'Dataset monitors (Preview)'. Under 'Create dataset', there are four options: 'From local files', 'From datastore', 'From web files' (which is also highlighted with a red box), and 'From Open Datasets'. Below these options, it says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampledata.blob.core.windows.net/data/flightdelays/flightdelays.csv>

6. Provide **flightdelays-automl** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there are tabs for 'Basic info', 'Settings and preview', 'Schema', and 'Confirm details'. The 'Basic info' tab is selected. It contains fields for 'Web URL' (with the value 'https://introtomlsampledata.blob.core.windows.net/data/flightdelays/flightdelays.csv'), 'Name' (with the value 'flightdelays-automl'), 'Dataset version' (set to 1), 'Dataset type' (set to 'Tabular'), and a 'Description' field. At the bottom, there are 'Back', 'Next', and 'Cancel' buttons, with 'Next' highlighted with a red box.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Review the dataset and then select **Next**

The screenshot shows the 'Create dataset from web files' wizard. The 'Settings and preview' step is active. The 'Column headers' dropdown is set to 'All files have same headers'. A red box highlights this dropdown. Below it is a preview table of flight data with several rows selected. The 'Next' button is highlighted with a blue box.

Task 3: Select Columns

1. Select columns from the dataset to include as part of your training data. Exclude the following columns: **Path, Month, Year, Timezone, Year_R, Timezone_R**, and then select **Next**

The screenshot shows the 'Create dataset from web files' wizard. The 'Schema' step is active. A scroll arrow points down to the bottom of the list. Several checkboxes for columns are highlighted with red boxes: Path, Month, and Year. A note 'Scroll to also exclude: Timezone, Year_R, and Timezone_R' is placed next to the scroll arrow. The 'Next' button is highlighted with a blue box.

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' blade in the Azure portal. A modal window titled 'Create dataset from web files' is open, showing the 'Confirm details' step. The 'Basic info' section contains the dataset name 'flightdelays-automl', version '1', and a tabular dataset type. The 'Web URL' field shows the URL 'https://introtomlsampledata.blob.core.windows.net/data/flightdelays/flightdelays.csv'. The 'File settings' section includes options for file format (Delimited), delimiter (Comma), encoding (UTF-8), column headers (All files have same headers), and skip rows (None). At the bottom of the modal, there is a 'Create' button, which is highlighted with a red box.

Exercise 2: Setup New Automated Machine Learning Experiment

Task 1: Create New Automated Machine Learning Experiment

1. From the studio home, select **Create new, Automated ML run**

The screenshot shows the Microsoft Azure Machine Learning studio home page. On the left, there is a navigation bar with icons for preview, workspace, and other tools. The main area has a 'Welcome to the studio!' message. Below it, there are four main cards: 'Notebooks' (code with Python SDK and run sample experiments), 'Automated ML' (automatically train and tune a model using a target metric), 'Designer' (drag-and-drop interface from prepping data to deploying models), and 'Tutorials' (train your first ML model with Notebook, what are compute targets in Azure Machine Learning?, and deploy models with Azure Machine Learning). On the far left, there is a sidebar with a 'Create new' dropdown menu. Under this menu, 'Automated ML run' is highlighted with a red box. Other options in the dropdown include 'Notebook', 'Pipeline', and 'Dataset'. At the bottom right of the main area, there is a link 'View all tutorials →'.

2. This will open a **Create a new automated machine learning experiment** page

Task 2: Select Training Data

1. Select the dataset **flightdelays-automl** and then select **Next**

Create a new Automated ML run

The screenshot shows the 'Select dataset' step of the 'Create a new Automated ML run' wizard. On the left, a vertical navigation bar lists three steps: 'Select dataset' (selected), 'Configure run', and 'Task type and settings'. The main area is titled 'Select dataset' with the sub-instruction: 'Select a dataset from the list below, or create a new dataset. Automated ML currently only supports tabular data for authoring runs.' Below this is a search bar with 'Create dataset' and 'Search to filter items...' options. A table lists datasets: 'flightdelays-automl' (selected and highlighted with a red border), 'flightdelays', 'Bike Rental Hourly', and 'nyc-taxi-sample-dataset'. The table columns are 'Dataset name', 'Dataset type', and 'Created'. The 'flightdelays-automl' row also includes a 'Feb' column.

Dataset name	Dataset type	Created
flightdelays-automl	Tabular	Feb
flightdelays	Tabular	Feb
Bike Rental Hourly	Tabular	Feb
nyc-taxi-sample-dataset	Tabular	Feb

Task 3: Create a new Automated ML run

1. Provide an experiment name: **flight-delay**
2. Select target column: **ArrDel15**
3. Select compute target: **select the available compute**
4. Select **Next**

The screenshot shows the 'Configure run' step of the 'Create a new Automated ML run' wizard. On the left, a vertical navigation bar lists three steps: 'Select dataset' (selected), 'Configure run' (selected), and 'Task type and settings'. The main area is titled 'Configure run' with the sub-instruction: 'Configure the experiment. Select from existing experiments or define a new name, select the target column and the training compute to use. Learn more on how to configure the experiment'. It shows a dataset selection section with 'flightdelays-automl' selected. Below it, the 'Experiment name' field is set to 'flight-delay' (highlighted with a red border) and the 'Target column' is set to 'ArrDel15' (highlighted with a red border). At the bottom, there are buttons for 'Back', 'Next' (highlighted with a red border), and 'Cancel'.

Task 4: Setup Task type and Settings

1. Select task type: **Classification**, and then select **View additional configuration settings**

The screenshot shows the 'Create a new Automated ML run' wizard. On the left, a vertical navigation bar lists steps: 'Select dataset', 'Configure run', and 'Task type and settings'. The 'Task type and settings' step is currently active. In the main panel, under 'Select task type', there are three options: 'Classification' (selected), 'Regression', and 'Time series forecasting'. Below each option is a brief description. At the bottom of the panel are 'Back', 'Finish', and 'Cancel' buttons. A red box highlights the 'Classification' section and the 'View additional configuration settings' link.

2. This will open the **Additional configurations** dialog.
3. Provide the following information and then select **Save**
 1. Primary metric: **AUC weighted**
 2. Exit criteria, Training job time (hours): **1**
 3. Exit criteria, Metric score threshold: **0.7**

The screenshot shows the 'Create a new Automated ML run' wizard with the 'Additional configurations' dialog open. The 'Primary metric' dropdown is set to 'AUC weighted'. Under 'Exit criterion', 'Training job time (hours)' is set to '1' and 'Metric score threshold' is set to '0.7'. The 'Save' button at the bottom right of the dialog is highlighted with a red box.

4. Note that we are setting a metric score threshold to limit the training time. In practice, for initial experiments, you will typically only set the training job time to allow AutoML to discover the best algorithm to use for your specific data.

Exercise 3: Start and Monitor Experiment

Task 1: Start Experiment

1. Select **Finish** to start running the experiment

The screenshot shows the 'Create a new Automated ML run' wizard. On the left, a vertical sidebar lists steps: 'Select dataset' (done), 'Configure run' (done), and 'Task type and settings'. The main area is titled 'Select task type' with the sub-instruction 'Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.' It contains three options: 'Classification' (selected, checked), 'Regression', and 'Time series forecasting'. Below these are two small links: 'View additional configuration settings' and 'View featurization settings'. At the bottom are 'Back', 'Finish' (highlighted with a red box), and 'Cancel' buttons.

Task 2: Monitor Experiment

1. The experiment will run for about 30 min. Note that most of the time will be spent in the data preparation step and once the data preparation is done, the experiment will take an additional 1-2 minutes to complete.
2. In the **Details** tab, observe the **run status** of the job.

The screenshot shows the 'Run Detail' page for Run ID 130, which is currently 'Preparing'. The top navigation bar shows 'learn-ws > Automated ML > Run Detail'. The sidebar has tabs: 'Details' (selected, highlighted with a red box), 'Models', 'Data guardrails', 'Properties', 'Logs', and 'Outputs'. The main content area is titled 'Run details' and includes sections for 'Task type' (Classification) and 'Primary metric' (AUC weighted). Below this, the 'Run status' is listed as 'Preparing' (also highlighted with a red box). Further down are 'Experiment name' (flight-delay) and 'Run ID' (AutoML_c1665111-ea8e-4e44-81f8-4ef4e1887d56).

3. Select the **Models** tab, and observe the various algorithms the AutoML is evaluating. You can also observe the corresponding **AUC weighted** scores for each algorithm.

Run 130 Running

Refresh Cancel

Details **Models** Data guardrails Properties Logs Outputs

Algorithm name AUC weighted ↓ Created Duration Status Model

MaxAbsScaler, LightGBM	0.6941101923654169	Feb 7, 2020 10:29 AM	00:01:59	Completed	Download
StandardScalerWrapper, XGBoostClassifier	NaN	Feb 7, 2020 10:31 AM		Running	

< Prev Next >

Note that we have set a metric score threshold to limit the training time. As a result you might see only one algorithm in your models list.

4. Select **Details** and wait till the run status becomes **Completed**.

Run 1 Completed

Refresh Cancel

Details Data guardrails Models Logs Outputs Child runs Snapshot

Properties

Status	Completed
Created	Jun 15, 2020 8:03 PM
Duration	2m 19.43s
Compute target	
Run ID	AutoML_75c97d70-18bc-46d9-8871-6ebad75ac03a
Run number	1
Script name	

Best model summary

Algorithm name	MaxAbsScaler, LightGBM
AUC weighted	0.73756 View all other metrics
Sampling	5% (i)
Registered models	No registration yet
Deploy status	No deployment yet

Run summary

5. While you wait for the model training to complete, you can learn to view and understand the charts and metrics for your automated machine learning run by selecting [Understand automated machine learning results](#).

Exercise 4: Review Best Model's Performance

Task 1: Review Best Model Performance

1. The **Details** tab shows the **Best model summary**. Next, select **Algorithm name** to review the model details.

The screenshot shows the Azure Machine Learning studio interface. On the left, there is a sidebar with various options like New, Home, Notebooks, Automated ML, Designer, Datasets, Experiments, Pipelines, Models, Endpoints, Compute, and Datastores. The 'Run 1' card is displayed, indicating it is completed. The 'Details' tab is selected. In the main pane, under 'Properties', there is a 'Best model summary' section which is highlighted with a red box. This section contains the algorithm name 'MaxAbsScaler, LightGBM'. Other details shown include Status (Completed), Created (Jun 15, 2020 8:03 PM), Duration (2m 19.43s), Compute target (qs-compute), and Run ID (AutoML_75c97d70-18bc-46d9-8871-6ebad75ac03a).

2. From the **Model details** tab, to view the various metrics to evaluate the best model performance, select **View all other metrics**.

Run 3 ✓ Completed

The screenshot shows the 'Model details' tab for Run 3. The 'Model' tab is selected. Under the 'Model summary' section, the algorithm name is listed as 'MaxAbsScaler, LightGBM'. The AUC weighted metric is shown as 0.73756, with a link to 'View all other metrics' which is highlighted with a red box. Other sections include Sampling (5%), Registered models (No registration yet), and Deploy status (No deployment yet).

-
3. Review the model performance metrics and then select **Close**.

Run Metrics

X

Accuracy

0.79083

AUC macro

0.73756

AUC micro

0.85635

AUC weighted

0.73756

Average precision score macro

0.69574

Average precision score micro

0.84495

Average precision score weighted

0.79893

Balanced accuracy

0.58215

F1 score macro

0.58813

F1 score micro

0.79083

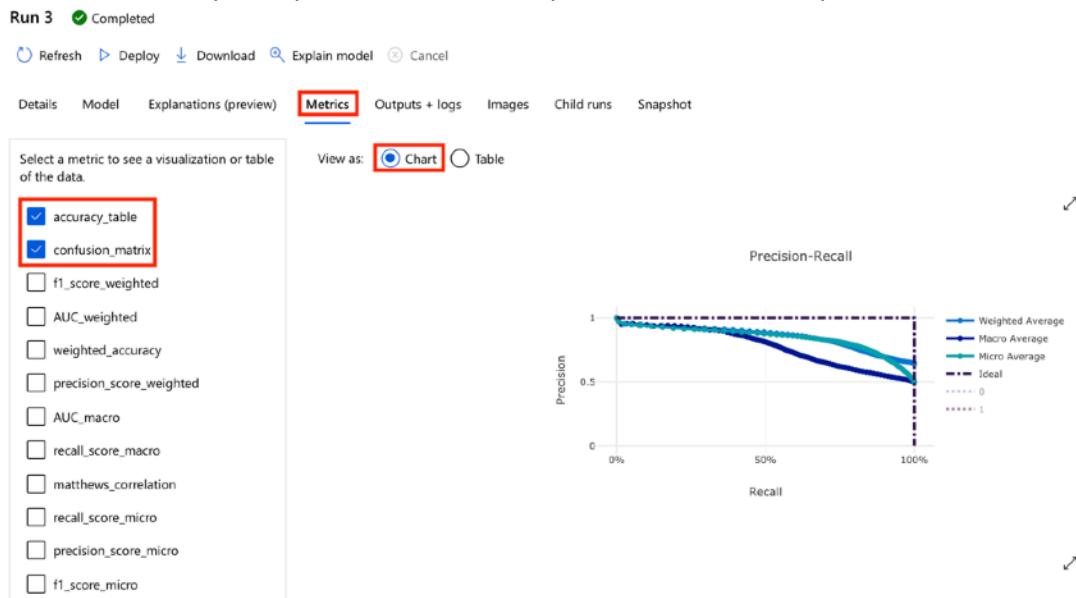
F1 score weighted

0.74266

Log loss

Close

4. Next, select **Metrics** to review the various model performance curves, such as Precision-Recall, ROC, Calibration curve, Gain & Lift curves, and Confusion matrix.



Next Steps

Congratulations! You have trained and evaluated your first automated machine learning model. 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.

Lab 8 - Two-Class Classifiers Performance

Overview

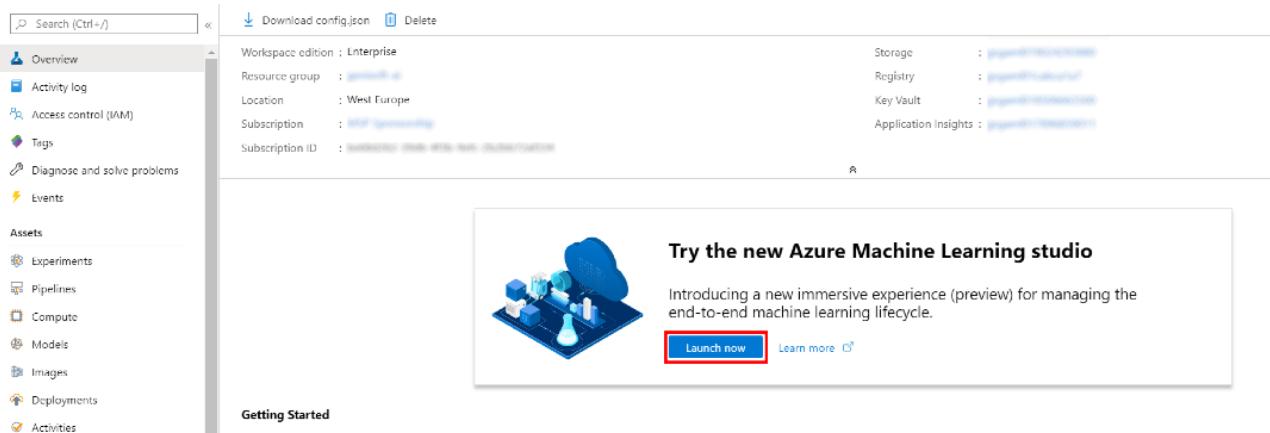
[Azure Machine Learning designer](#) (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two binary classifiers: Two-Class Boosted Decision Tree and Two-Class Logistic Regression for predicting customer churn. The goal is to run an expensive marketing campaign for high risk customers; thus, the **precision** metric is going to be key in evaluating performance of these two algorithms. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Create Training Pipeline

Task 1: Open Sample 5: Binary Classification – Customer Relationship Prediction

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



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124southcentralus

Get started

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**.

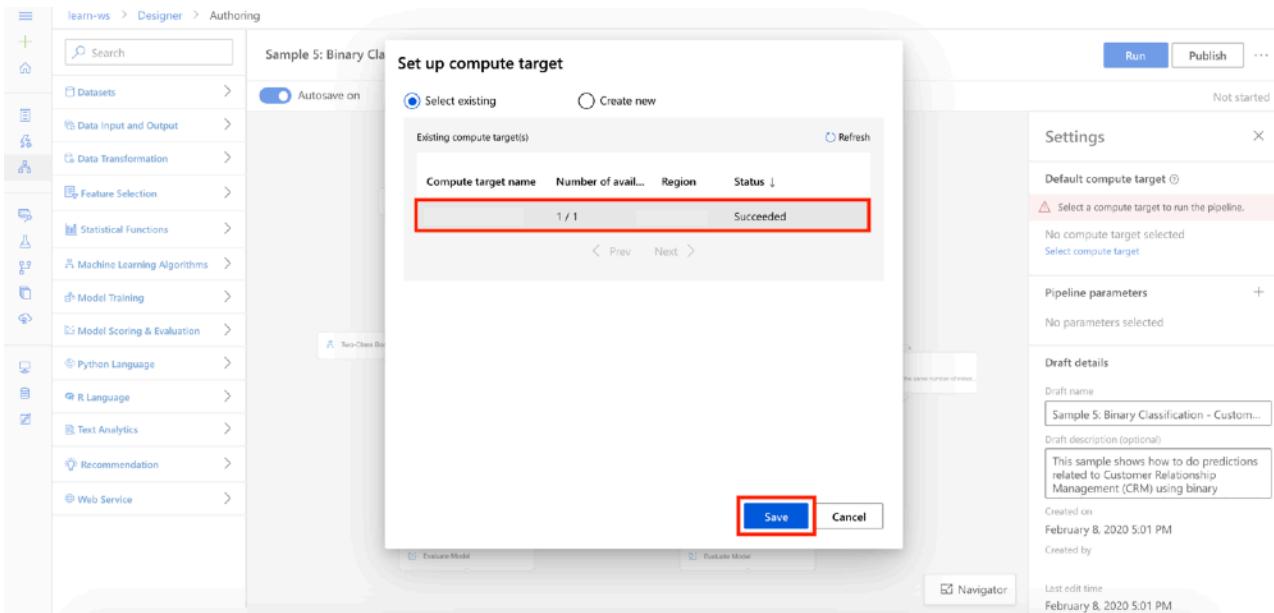
4. From the studio, select **Designer, Sample 5: Binary Classification – Customer Relationship Prediction.**

Task 2: Setup Compute Target

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

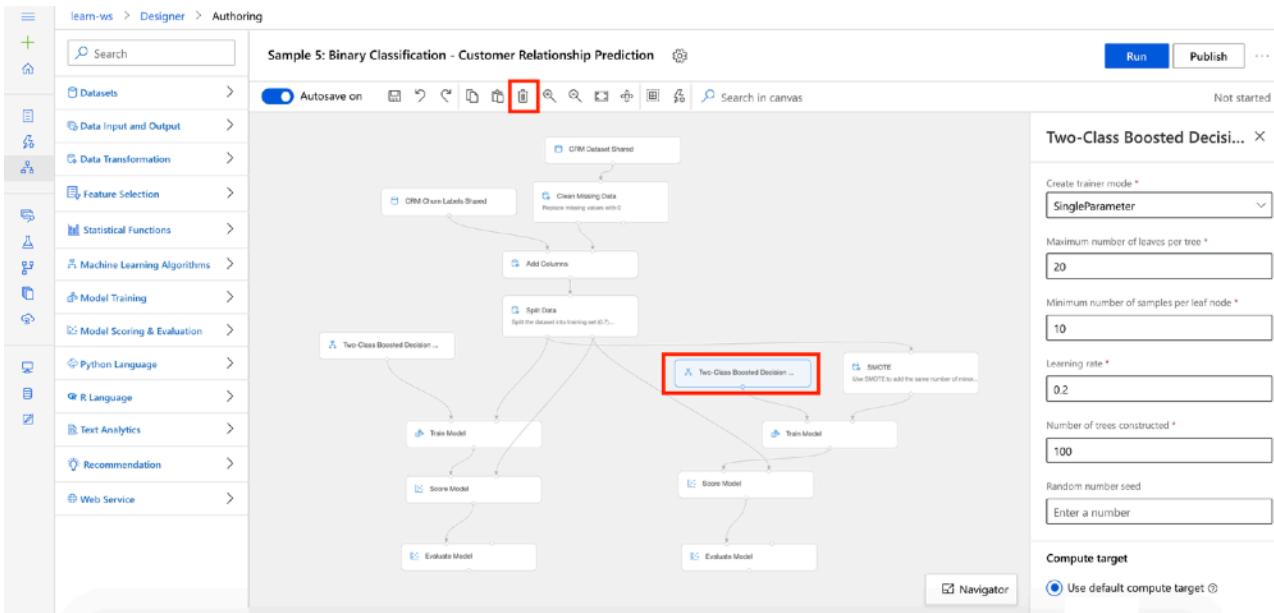
- In the **Set up compute target** editor, select the available compute, and then select **Save**.

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.

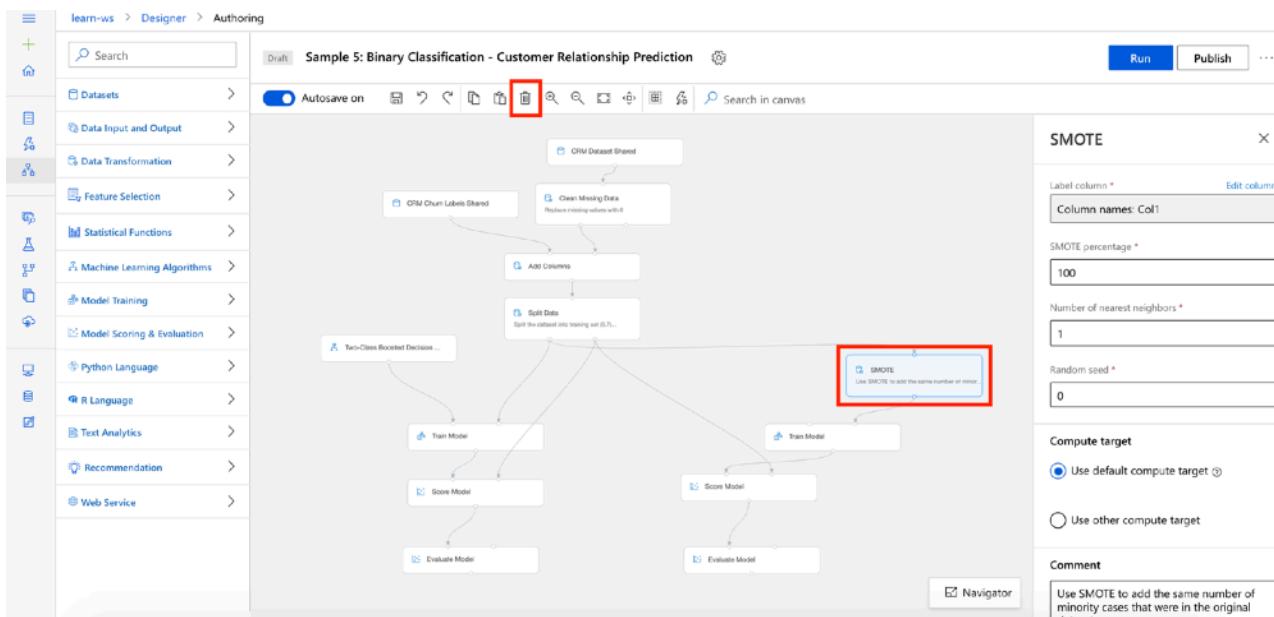


Task 3: Delete Pipeline Modules

- From the **right-hand-side** of the pipeline, select the **Two-Class Boosted Decision Tree module** and then select the **Delete Icon**.

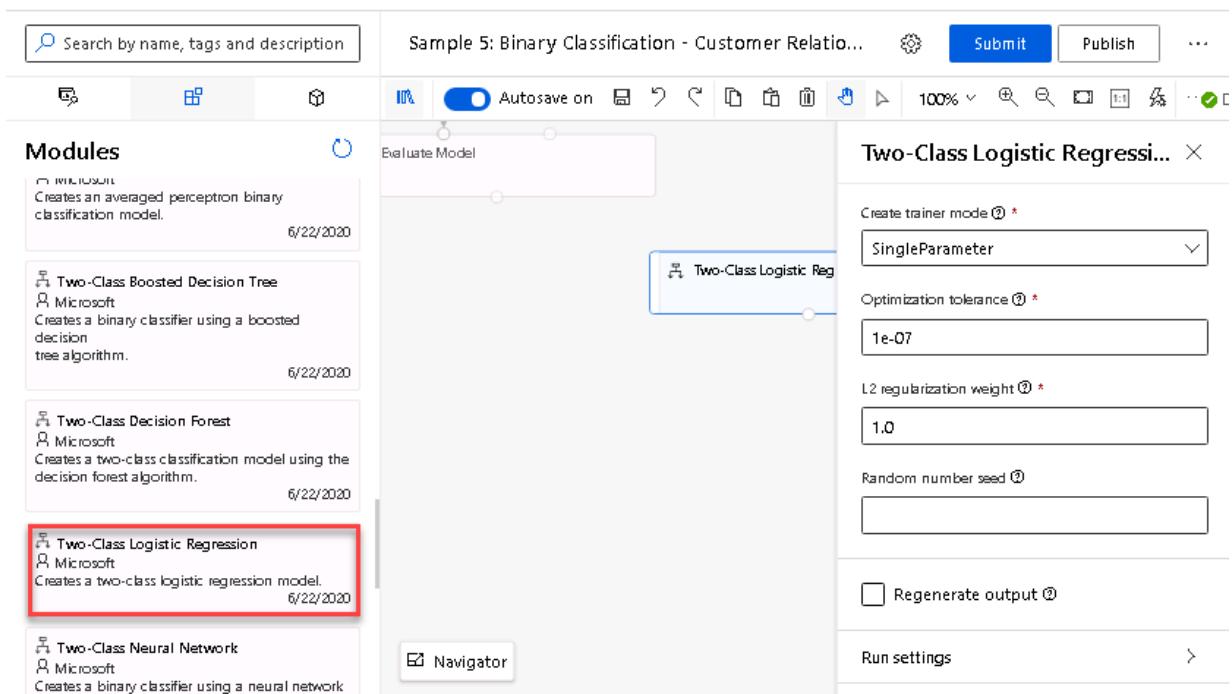


2. From the **right-hand-side** of the pipeline, select the **SMOTE module** and then select the **Delete Icon**.



Task 4: Setup the Two-Class Logistic Regression Module

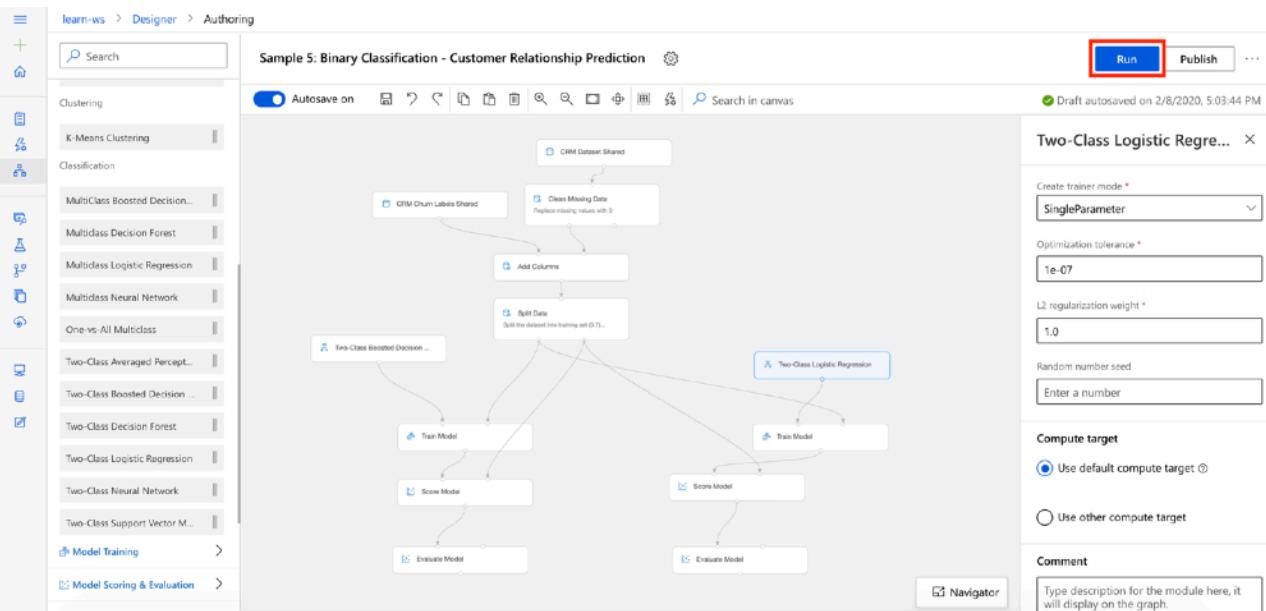
1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Two-Class Logistic Regression** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Two-Class Logistic Regression** module to the first input of the **Train Model** module
 4. Connect the first output of the **Split Data** module to the second input of the **Train Model** module



Exercise 2: Submit Training Pipeline

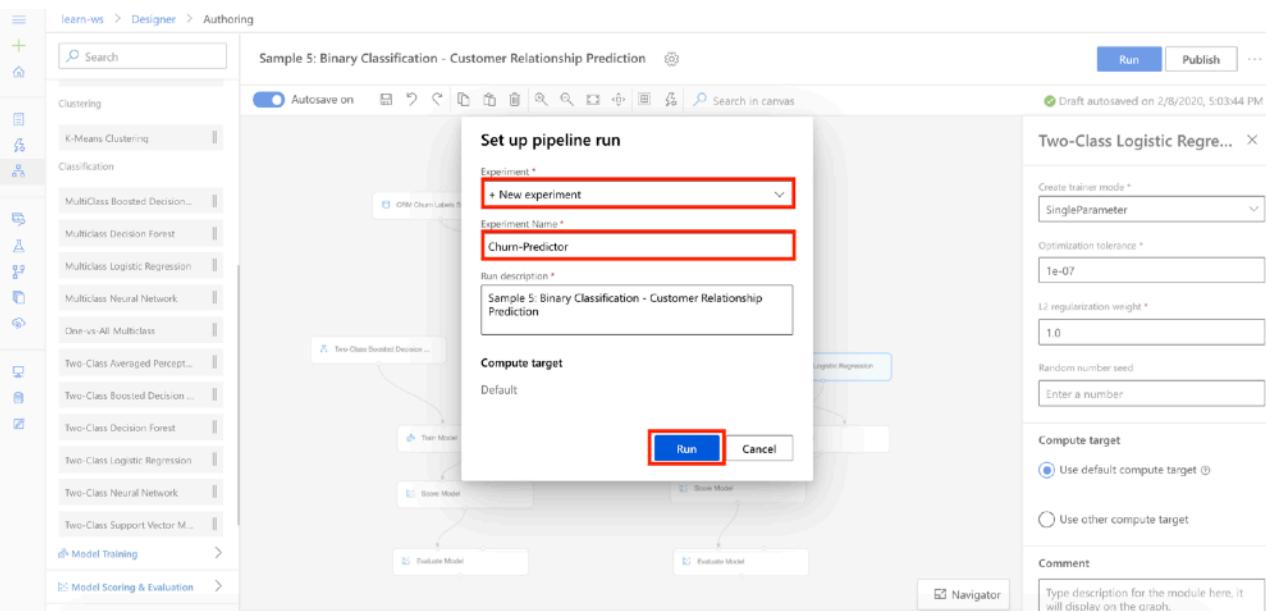
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



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

2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: Churn-Predictor**, and then select **Submit**.

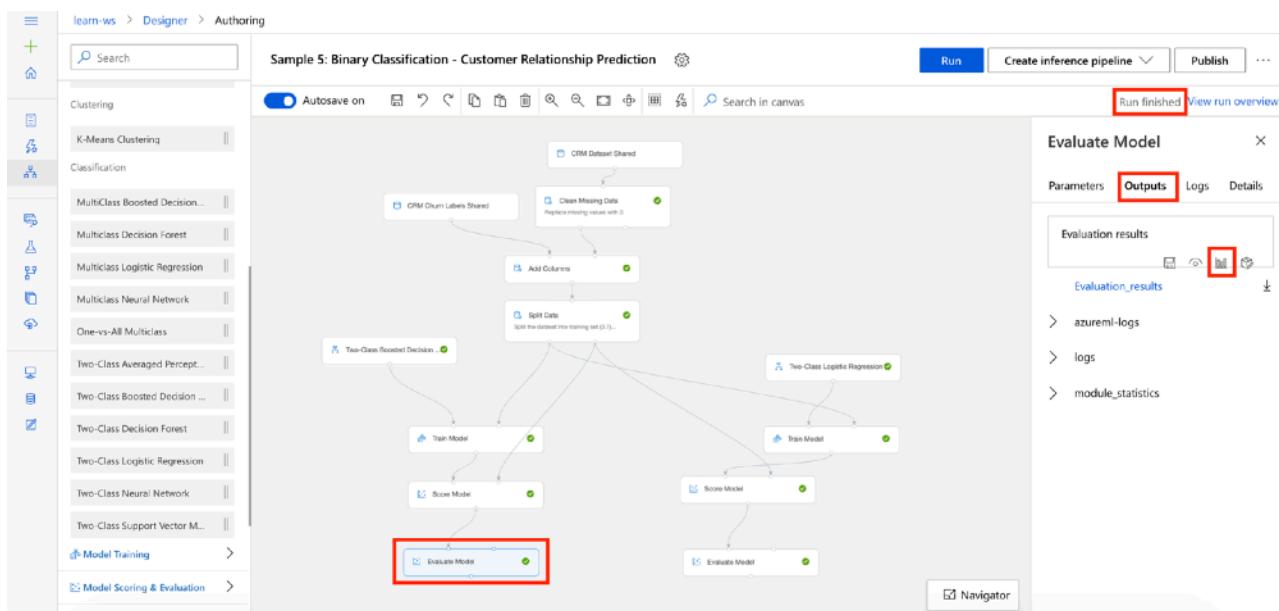


3. Wait for pipeline run to complete. It will take around **5 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the evaluation metrics for the classification algorithm used in this lab by selecting [Metrics for classification models](#).

Exercise 3: Compare Model Performance

Task 1: Open Evaluation Results for Two-Class Boosted Decision Tree

- From the left-hand-side of the pipeline, select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog for the **Two-Class Boosted Decision Tree** module.



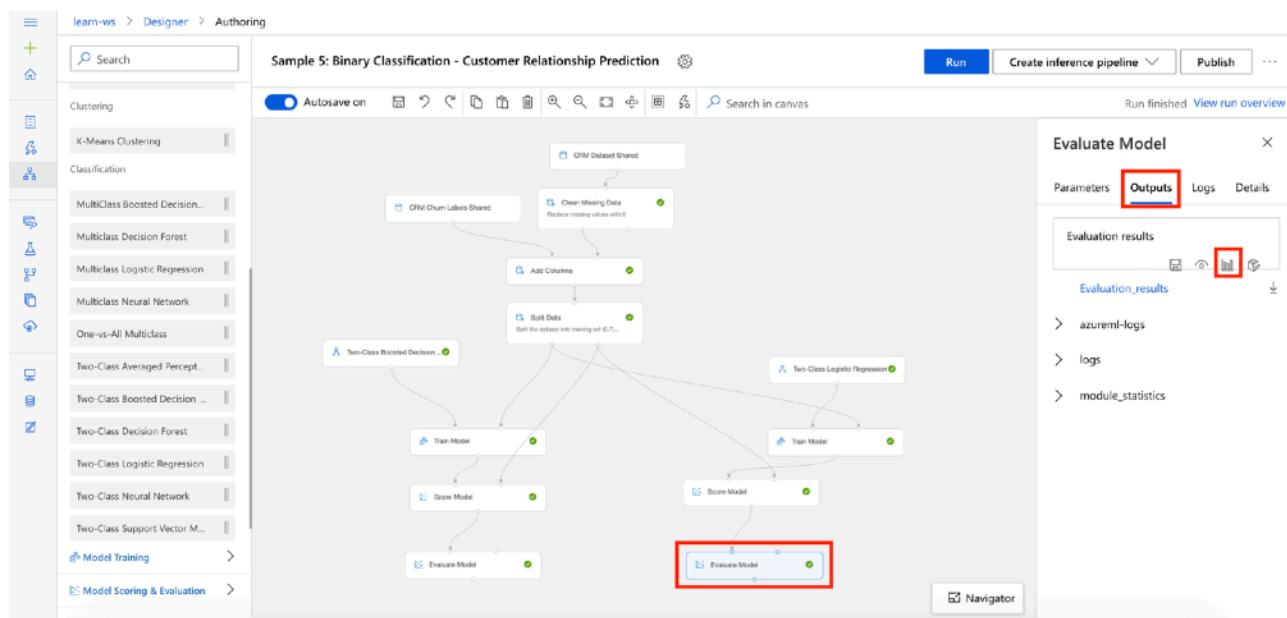
Task 2: Evaluate Two-Class Boosted Decision Tree Performance

- Scroll down to review model performance metrics for **Two-Class Boosted Decision Tree**. Observe that the **Precision** value is around **0.7**.



Task 3: Open Evaluation Results for Two-Class Logistic Regression

- From the right-hand-side of the pipeline, select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog for the **Two-Class Logistic Regression** module.



Task 4: Evaluate Two-Class Logistic Regression Performance

- Scroll down to review model performance metrics for **Two-Class Logistic Regression**. Observe that the **Precision** value is around **0.3**.



Task 5: Conclusion

1. Based on the primary performance metric, **Precision**, it shows that the **Two-Class Boosted Decision Tree** algorithm outperforms the **Two-Class Logistic Regression** algorithm.

Next Steps

Congratulations! You have trained and compared performance of two different classification machine learning models. 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.

Lab 9 - Multi-Class Classifiers Performance

Overview

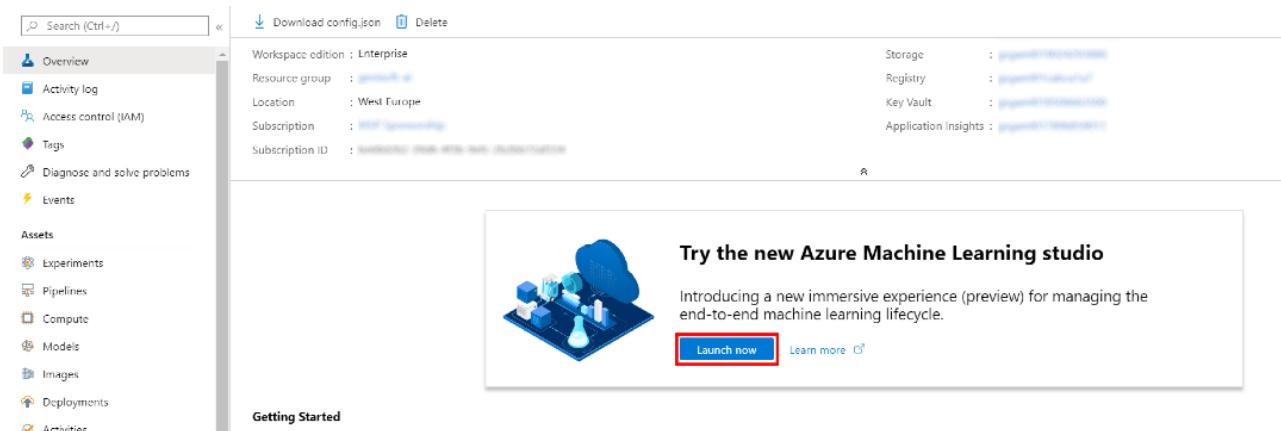
Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two different multiclass classification approaches: Two-Class Support Vector Machine used with One-vs-All Multiclass module vs Multiclass Decision Forest. We will apply the two approaches for the letter recognition problem and compare their performance. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Create Training Pipeline

Task 1: Open Sample 12: Multiclass Classification - Letter Recognition

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

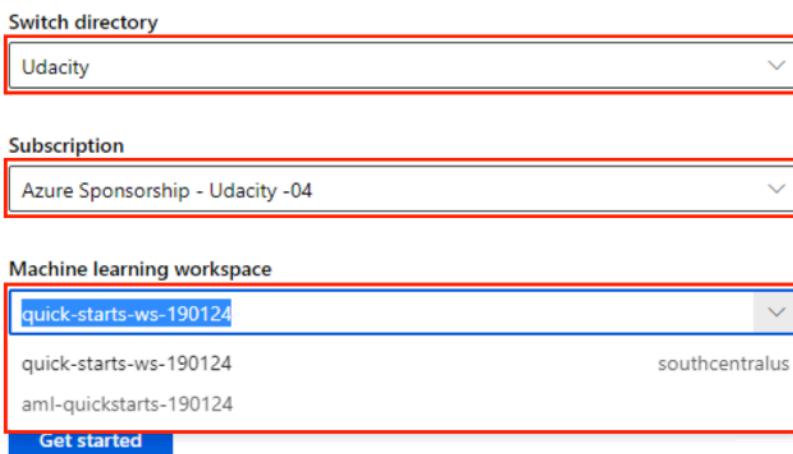


The screenshot shows the Azure Machine Learning workspace overview page. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets (Experiments, Pipelines, Compute, Models, Images, Deployments, Activities), and Getting Started. The main area displays workspace details: Edition: Enterprise, Resource group: [general-ml](#), Location: West Europe, Subscription: [MLT Sponsorship](#), Subscription ID: [SUBSCRIPTION_ID](#). It also lists Storage, Registry, Key Vault, and Application Insights. Below this is a 'Getting Started' section with a 'Try the new Azure Machine Learning studio' card. The card features a 3D cityscape icon on a blue base, the text 'Try the new Azure Machine Learning studio', 'Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle.', and a 'Launch now' button which is highlighted with a red box. There's also a 'Learn more' link.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



The screenshot shows the 'Welcome to the studio!' configuration screen. It has three dropdown menus:

- Switch directory:** Set to 'Udacity'.
- Subscription:** Set to 'Azure Sponsorship - Udacity -04'.
- Machine learning workspace:** Shows a list of workspaces:
 - 'quick-starts-ws-190124' (selected)
 - 'quick-starts-ws-190124'
 - 'aml-quickstarts-190124'

A 'Get started' button is located at the bottom of the form.

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**.

4. From the studio, select **Designer**, **Show more samples**.

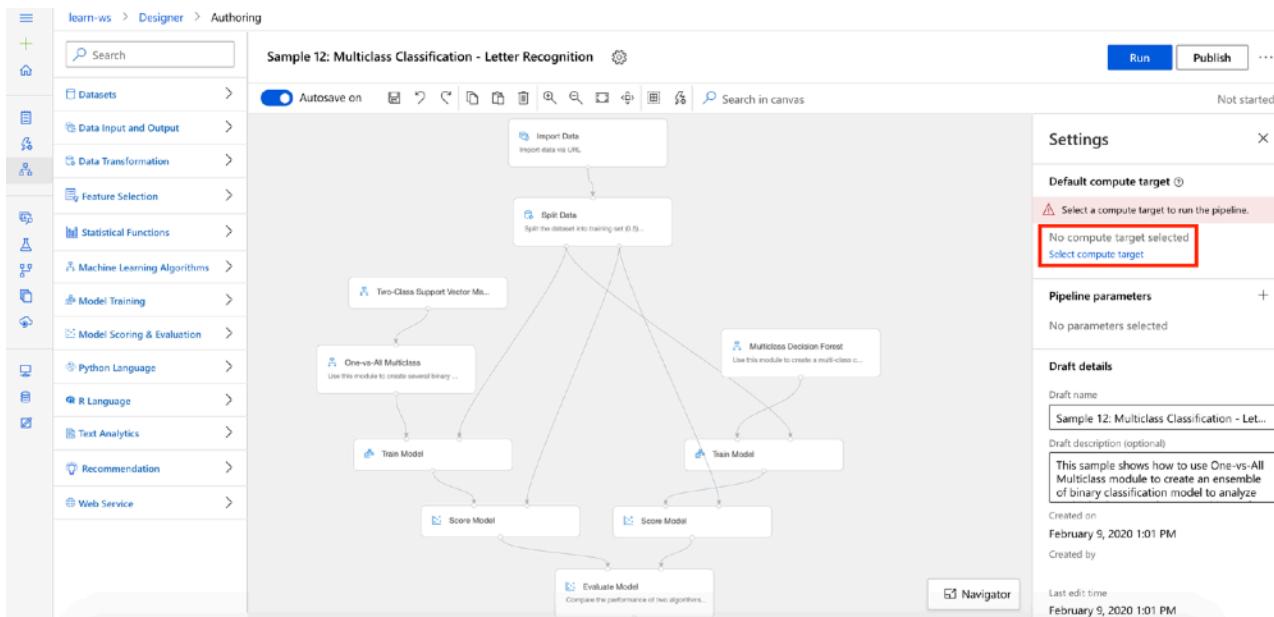
The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, there's a sidebar with icons for Home, New pipeline, Pipeline drafts, Pipeline runs, Refresh, Delete, and a search bar labeled 'Search to filter items...'. The main area is titled 'Designer' and has sections for 'New pipeline' and 'Pipelines'. Under 'New pipeline', there are five sample pipelines: 'Easy-to-use prebuilt modules' (with a plus icon), 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', 'Sample 4: Binary Classification with custom Python script - ...', and 'Sample 5: Binary Classification - Customer Relationship Pred...'. A red box highlights the 'Show more samples' button in the top right corner. Under 'Pipelines', it says 'Pipeline drafts' and 'Pipeline runs', followed by a message 'No pipeline drafts to display'.

5. Select **Sample 12: Multiclass Classification - Letter Recognition**.

This screenshot shows the same Designer interface as above, but with a red box highlighting 'Sample 12: Multiclass Classification - Letter Recognition'. This sample is represented by a thumbnail showing three letters (A, B, C) in boxes and the text 'ABC'. Below the thumbnail, it says 'Sample 12: Multiclass Classification - Letter Recog...'. The rest of the interface and sample list are identical to the previous screenshot.

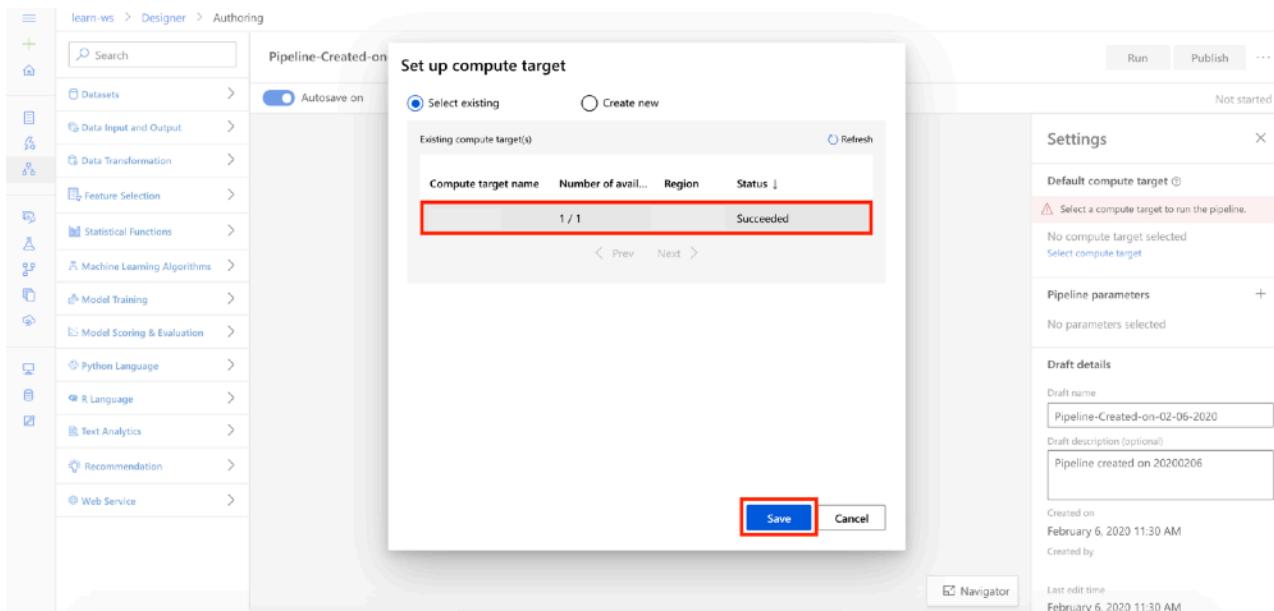
Task 2: Setup Compute Target

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



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

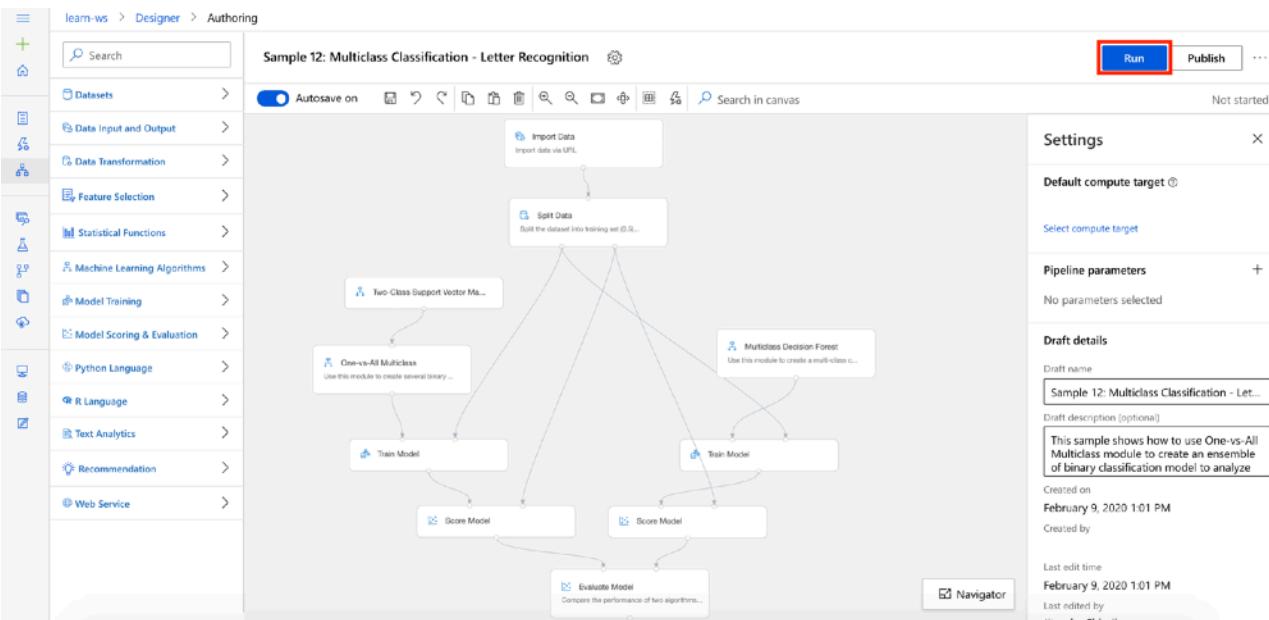
Note: If you are facing difficulties in accessing pop-up windows or buttons in the user interface, please refer to the Help section in the lab environment.



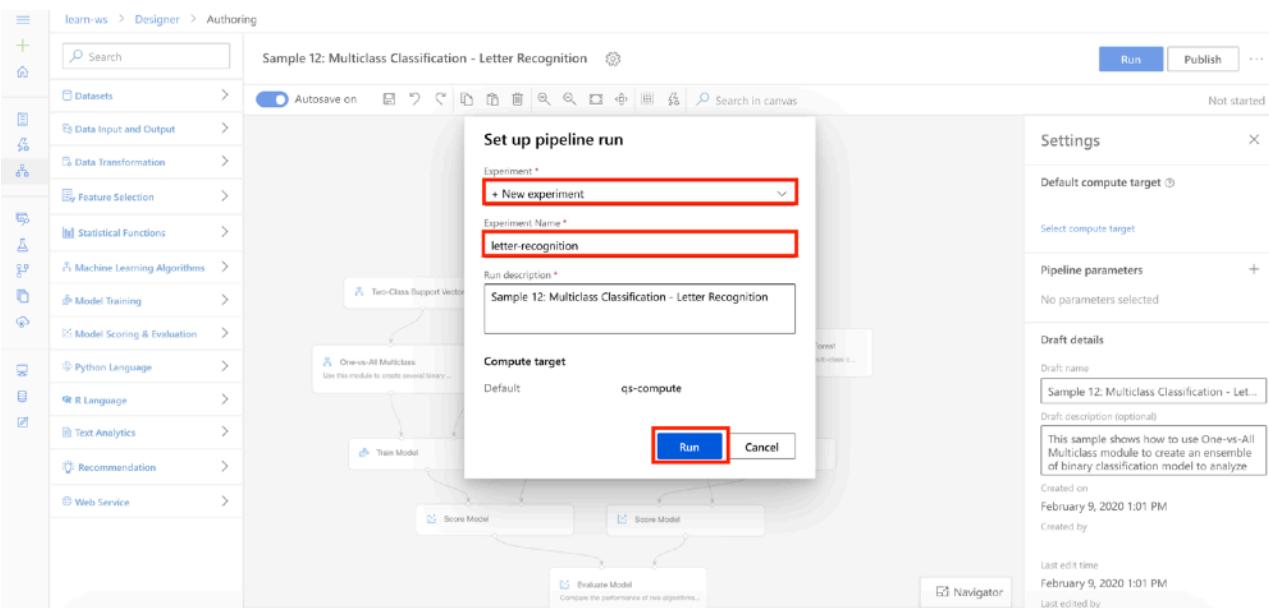
Exercise 2: Submit Training Pipeline

Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



2. Please note that the button name in the UI is changed from **Run** to **Submit**.
3. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: letter-recognition**, and then select **Submit**.

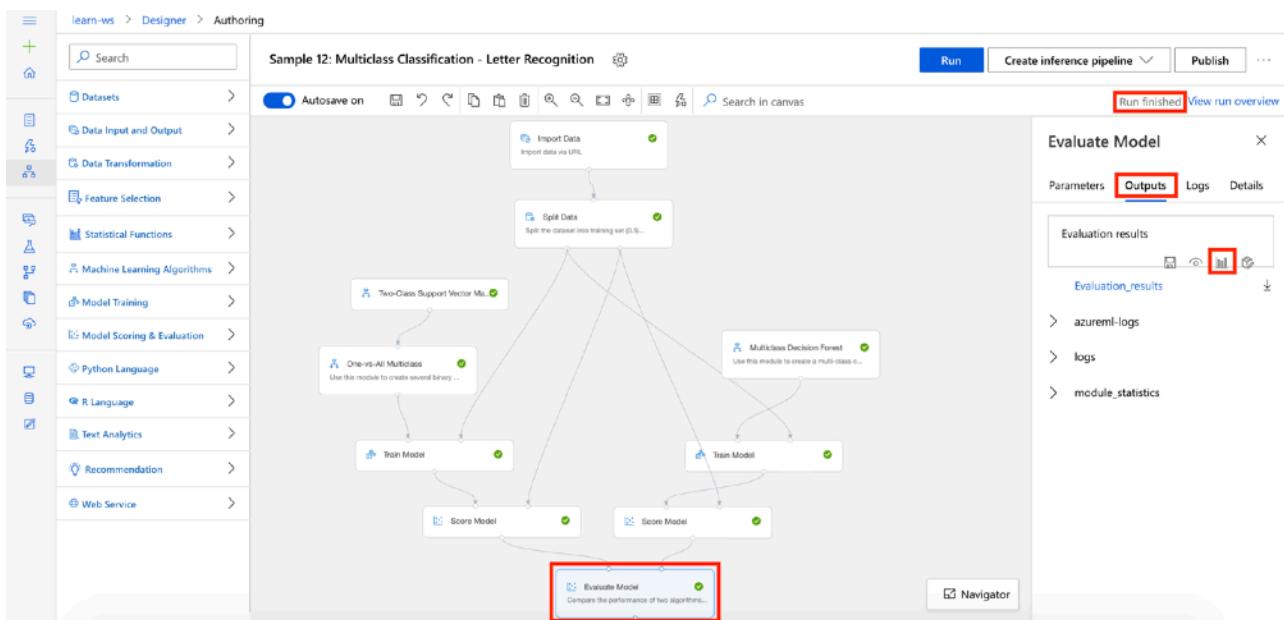


4. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.
5. While you wait for the model training to complete, you can learn more about the **One-vs-All Multiclass** module used in this lab by selecting **One-vs-All Multiclass**.

Exercise 3: Compare Model Performance

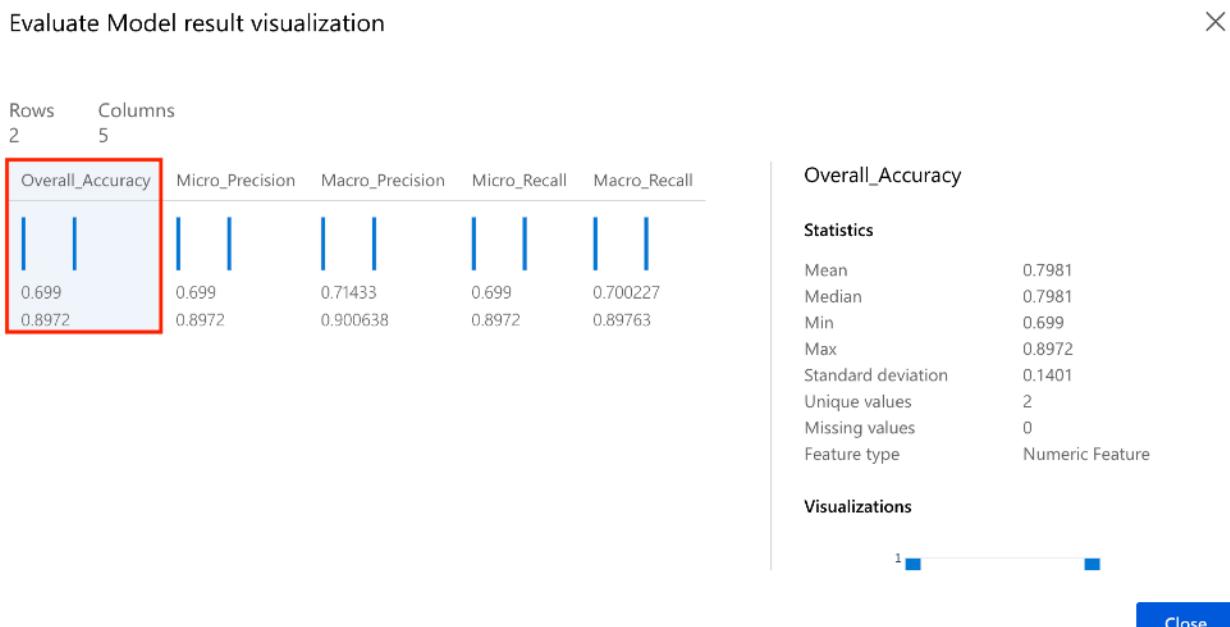
Task 1: Open Evaluation Results

- Select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog.



Task 2: Compare Performance Metrics

- Select the regression performance metric **Overall_Accuracy** and compare performance of the two algorithms: **Two-Class Support Vector Machine** and **Multiclass Decision Forest**.



Task 3: Conclusion

1. The **Two-Class Support Vector Machine** algorithm is extended for multiclass classification problem by using the **One-vs-All Multiclass** module.
2. As you can observe that the native multiclass algorithm **Multiclass Decision Forest** outperforms the **Two-Class Support Vector Machine** across all key performance metrics.
3. One recommendation for next steps is to increase the **Number of iterations** parameter for the **Two-Class Support Vector Machine** module to an higher value like **100** and observe its impact on the performance metrics.

Next Steps

Congratulations! You have trained and compared performance of two different multiclass classification machine learning models. 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.

Lab 10 - Train a Classifier Using Automated Machine Learning

Overview

Automated machine learning picks an algorithm and hyperparameters for you and generates a model ready for deployment. There are several options that you can use to configure automated machine learning experiments.

Configuration options available in automated machine learning:

- Select your experiment type: Classification, Regression or Time Series Forecasting
- Data source, formats, and fetch data
- Choose your compute target
- Automated machine learning experiment settings
- Run an automated machine learning experiment
- Explore model metrics
- Register and deploy model

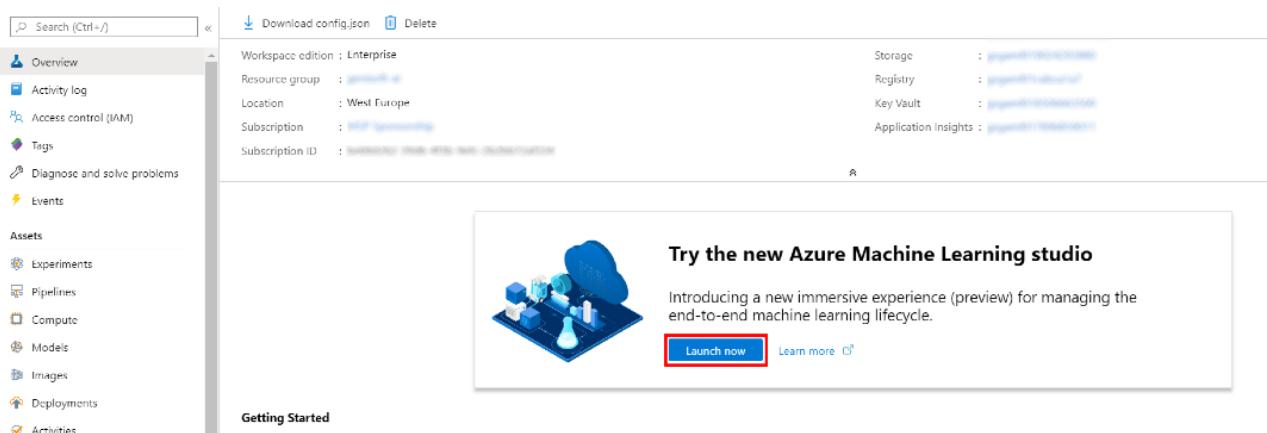
You can create and run automated machine learning experiments in code using the [Azure ML Python SDK](#) or if you prefer a no code experience, you can also create your automated machine learning experiments in [Azure Machine Learning Studio](#).

In this lab, we will use Automated Machine Learning to find the best performing binary classification model for predicting customer churn. We will do all of this from the [Azure Machine Learning Studio](#) without writing a single line of code.

Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

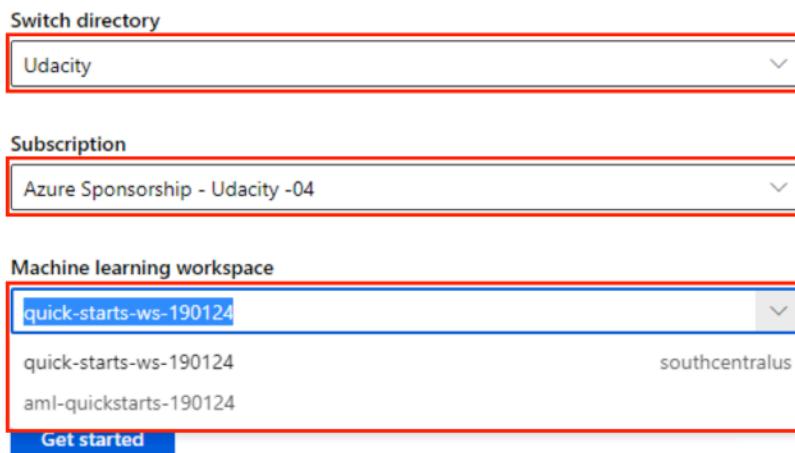
1. In [Azure portal](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



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**.

4. From the studio, select **Datasets**, + **Create dataset**, **From web files**. This will open the **Create dataset from web files** dialog on the right.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampled.blob.core.windows.net/data/crm-churn/crm-churn.csv>

6. Provide **CRM-Churn** as the Name, leave the remaining values at their defaults and select **Next**.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Review the dataset and then select **Next**

The screenshot shows the 'Create dataset from web files' wizard in the Azure portal. The current step is 'Settings and preview'. A red box highlights the 'Column headers' dropdown, which contains the option 'All files have same headers'. The 'Next' button at the bottom of the form is also highlighted with a red box.

Task 3: Select Columns

- Keep the default selections, and select **Next**

The screenshot shows the 'Create dataset from web files' wizard in the Azure portal. The current step is 'Schema'. A table lists the columns: Col1, Var6, Var7, Var13, Var21, Var22, Var24, Var25, Var28, Var38, Var57, Var65, and Var73. Each column has a blue checkbox next to it under the 'Include' column. The 'Next' button at the bottom of the form is highlighted with a red box.

Include	Column name	Properties	Type
<input checked="" type="checkbox"/>	Path	Not applicable to selecte...	String
<input checked="" type="checkbox"/>	Col1	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var6	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var7	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var13	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var21	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var22	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var24	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var25	Not applicable to selecte...	Integer
<input checked="" type="checkbox"/>	Var28	Not applicable to selecte...	Decimal

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' blade on the left and the 'Create dataset from web files' wizard on the right. The wizard has five steps: Basic info, Settings and preview, Schema, Confirm details, and Profile this dataset after creation. The 'Confirm details' step is active. In the 'Basic info' section, the name is set to 'CRM-Churn'. The 'File settings' section includes options for file format (Delimited), delimiter (Comma), encoding (UTF-8), column headers (All files have same headers), and skip rows (None). The 'Create' button at the bottom right is highlighted with a red box.

Exercise 2: Setup New Automated Machine Learning Experiment

Task 1: Create New Automated Machine Learning Experiment

1. From the studio home, select **Create new, Automated ML run**

The screenshot shows the Azure Machine Learning studio home page. On the left, there's a sidebar with icons for preview, workspace, and various services like notebooks, datasets, pipelines, and automated machine learning. A 'Create new' dropdown menu is open, showing 'Notebook', 'Pipeline', 'Dataset', and 'Automated ML run'. The 'Automated ML run' option is highlighted with a red box. To the right, there are several cards: 'Notebooks' (code with Python SDK and run sample experiments), 'Automated ML' (automatically train and tune a model using a target metric), 'Designer' (drag-and-drop interface for prepping data to deploying models), and two smaller cards below them: 'Train your first ML model with Notebook' and 'What are compute targets in Azure Machine Learning?'. At the bottom right, there's a link 'View all tutorials →'.

2. This will open a **Create a new automated machine learning experiment** page

Task 2: Select Training Data

1. Select the dataset **CRM-Churn** and then select **Next**

The screenshot shows the 'Create a new Automated ML run' wizard at the 'Select dataset' step. On the left, a sidebar lists 'Select dataset', 'Configure run', and 'Task type and settings'. The main area is titled 'Select dataset' with the sub-instruction 'Select a dataset from the list below, or create a new dataset. Automated ML currently only supports tabular data for authoring runs.' A table lists datasets: 'CRM-Churn' (selected, highlighted with a red border), 'Tabular', 'Created on Feb 8, 2020 7:08 PM', and 'Modified Feb 8, 2020 7:08 PM'. Below the table are navigation buttons '< Prev' and 'Next >'. At the bottom are 'Back', 'Next' (highlighted with a red border), and 'Cancel' buttons.

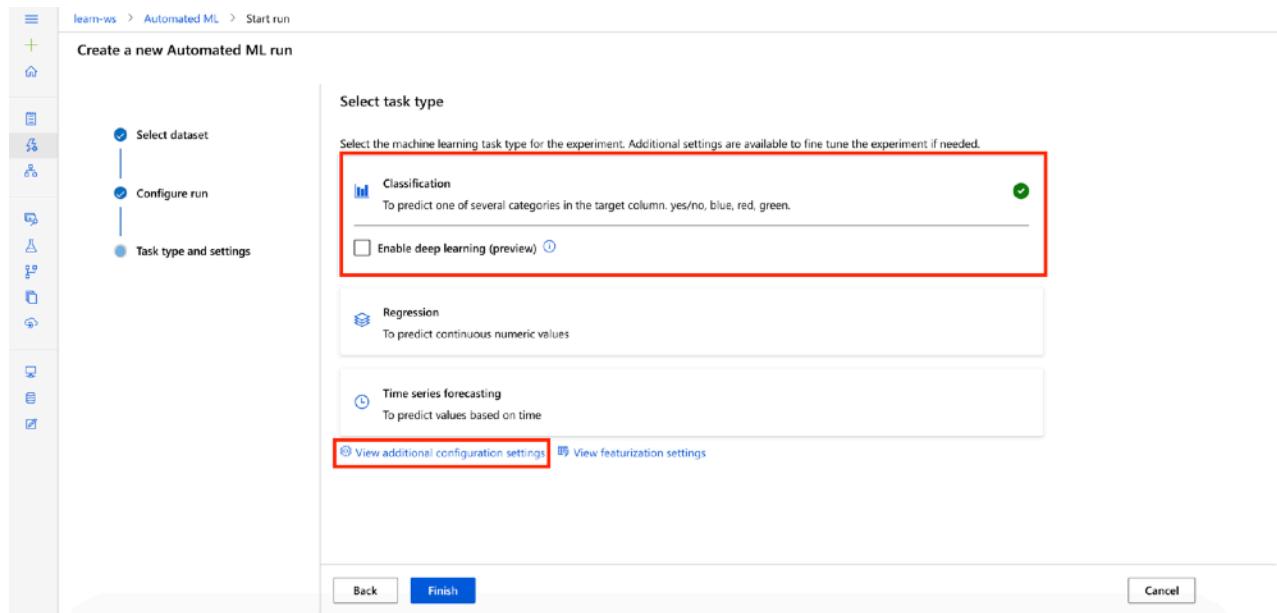
Task 3: Create a new Automated ML run

1. Provide an experiment name: **Churn-Predictor**
2. Select target column: **Col1**
3. Select compute target: **select the available compute**
4. Select **Next**

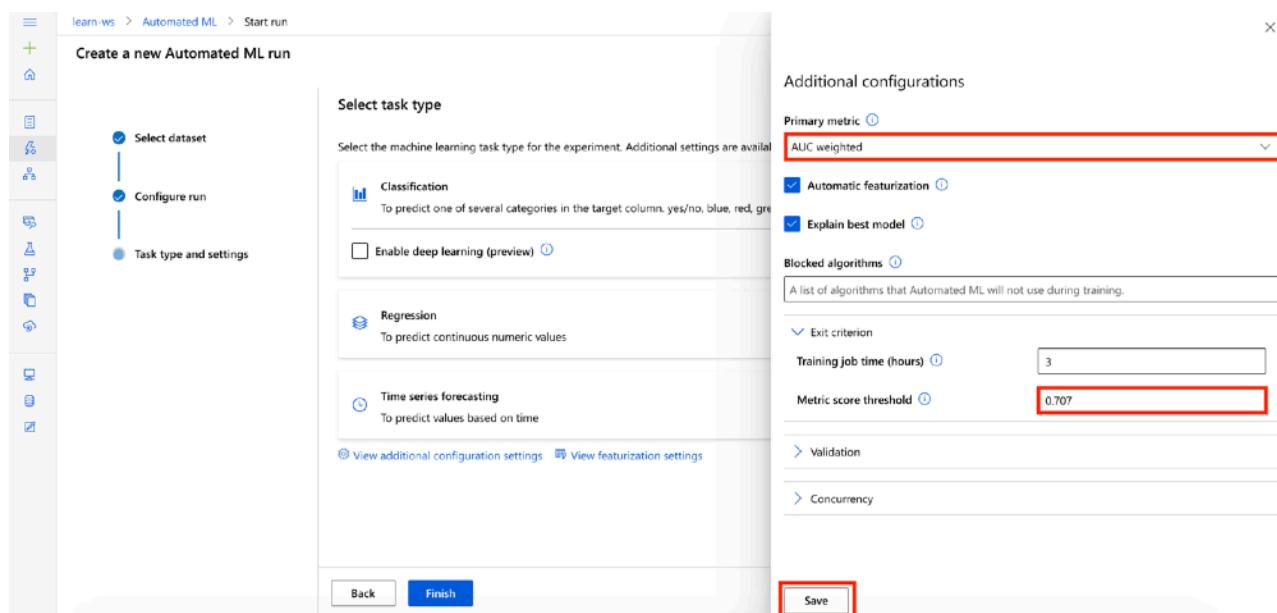
The screenshot shows the 'Create a new Automated ML run' wizard at the 'Configure run' step. On the left, a sidebar lists 'Select dataset' (selected), 'Configure run' (highlighted with a blue border), and 'Task type and settings'. The main area is titled 'Configure run' with the sub-instruction 'Configure the experiment: Select from existing experiments or define a new name, select the target column and the training compute to use. Learn more on how to configure the experiment'. It shows 'Dataset CRM-Churn (View dataset)'. The 'Experiment name *' field contains 'Churn-Predictor' (highlighted with a red border). The 'Target column *' dropdown contains 'Col1' (highlighted with a red border). The 'Select training compute target *' dropdown is empty (highlighted with a red border). Below these fields are 'Create a new compute' and 'Refresh compute' buttons. At the bottom are 'Back', 'Next' (highlighted with a red border), and 'Cancel' buttons.

Task 4: Setup Task type and Settings

1. Select task type: **Classification**, and then select **View additional configuration settings**



2. This will open the **Additional configurations** dialog.
3. Provide the following information and then select **Save**
 1. Primary metric: **AUC weighted**
 2. Exit criteria, Metric score threshold: **0.707**

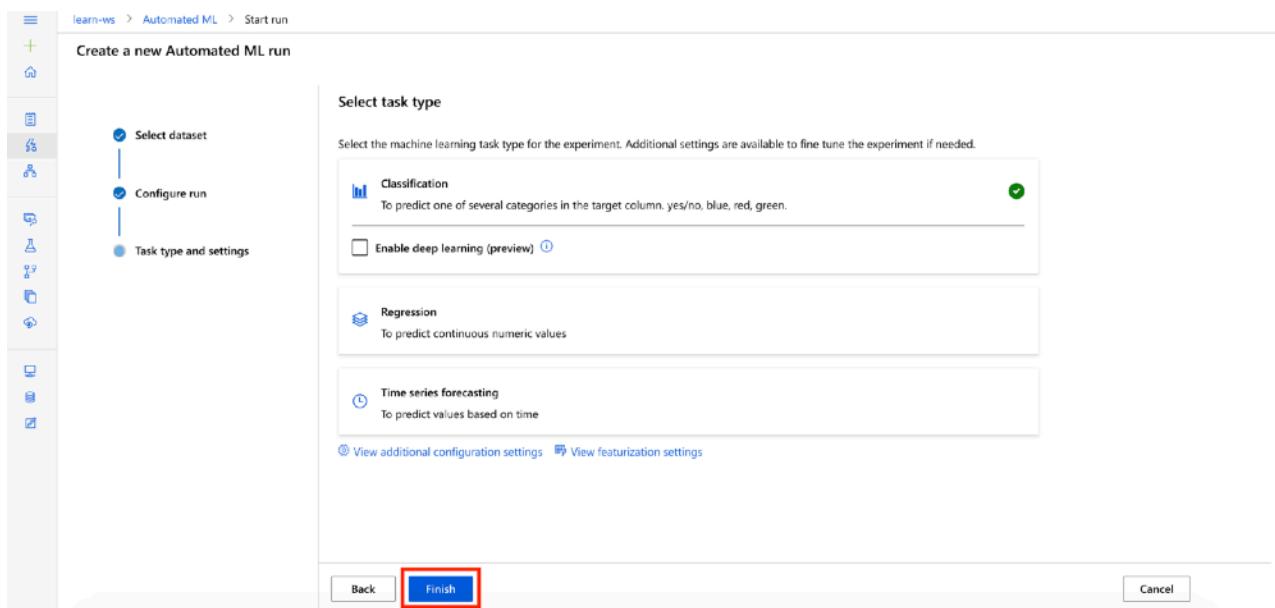


4. Note that we are setting a metric score threshold to limit the training time. In practice, for initial experiments, you will typically only set the training job time to allow AutoML to discover the best algorithm to use for your specific data.

Exercise 3: Start and Monitor Experiment

Task 1: Start Experiment

1. Select **Finish** to start running the experiment



Task 2: Monitor Experiment

1. The experiment will run for about 5 min
2. In the **Details** tab, observe the **run status** of the job.

The screenshot shows the 'Run 1' details page. The top navigation bar includes 'Run 1' (highlighted with a red box), 'Running', 'Refresh', and 'Cancel'. Below this, there are tabs: 'Details' (highlighted with a red box), 'Data guardrails', 'Models', 'Outputs + Logs', 'Child runs', and 'Snapshot'. The 'Properties' section contains fields like 'Status' (highlighted with a red box and set to 'Running'), 'Created' (Jun 28, 2020 2:54 PM), 'Compute target' (aml-compute), 'Run ID' (AutoML_09421b55-96a9-40c8-a3ae-32f129402d4a), 'Run number' (1), 'Script name' (--), 'Created by' (ODL_User 4977), 'Input datasets' (input_name: input_data, ID: 8dbbb96fc-3002-44e1-b921-8990072e3c58), 'Output datasets' (None), and 'Arguments' (None). The 'Run summary' section on the right shows 'Task type: Classification', 'Primary metric: AUC weighted', 'Run status: Running', and 'Experiment name: Churn-Predictor'.

3. Wait till the run status becomes **Completed**.

Run 1 Completed

Refresh Cancel

Properties

- Status Completed
- Created Jun 28, 2020 2:54 PM
- Duration 1m 49.42s
- Compute target aml-compute
- Run ID AutoML_09421b55-96a9-40c8-a3ae-32f129402d4a
- Run number 1
- Script name --
- Created by ODL_User 4977
- Input datasets Input name: input_data, ID: [Bdbbb968c-3002-44e1-b921-8990372d958](#)
- Output datasets None
- Arguments None
- [See all properties](#)
- [Raw JSON](#)

Best model summary

- Algorithm name **MaxAbsScaler, XGBoostClassifier**
- AUC weighted 0.70930 [View all other metrics](#)
- Sampling 100% ()
- Registered models No registration yet
- Deploy status No deployment yet

Run summary

- Task type Classification [View all run settings](#)
- Primary metric AUC weighted
- Run status Completed
- Experiment name Churn-Predictor

4. While you wait for the model training to complete, you can learn to view and understand the charts and metrics for your automated machine learning run by selecting [Understand automated machine learning classification results](#).

Exercise 4: Review Best Model's Performance

Task 1: Review Best Model Performance

1. From the **Details** tab review the best model's **Algorithm name** and its corresponding **AUC weighted** score. Next, select the best model's **Algorithm name**

Run 1 Completed

Refresh Cancel

Properties

- Status Completed
- Created Jun 28, 2020 2:54 PM
- Duration 1m 49.42s
- Compute target aml-compute
- Run ID AutoML_09421b55-96a9-40c8-a3ae-32f129402d4a
- Run number 1
- Script name --
- Created by ODL_User 4977
- Input datasets Input name: input_data, ID: [Bdbbb968c-3002-44e1-b921-8990372d958](#)
- Output datasets None
- Arguments None
- [See all properties](#)
- [Raw JSON](#)

Best model summary

- Algorithm name** MaxAbsScaler, XGBoostClassifier
- AUC weighted 0.70930 [View all other metrics](#)
- Sampling 100% ()
- Registered models No registration yet
- Deploy status No deployment yet

Run summary

- Task type Classification [View all run settings](#)
- Primary metric AUC weighted
- Run status Completed
- Experiment name Churn-Predictor

2. Select **View all other metrics** to review the various **Run Metrics** to evaluate the model performance. Next, select **Metrics**

The screenshot shows the Azure Machine Learning studio interface. On the left, there's a sidebar with various icons. The main area shows a 'Run 4' card with the status 'Completed'. Below the card, there are tabs: 'Details', 'Model' (which is selected and highlighted with a red box), 'Explanations (preview)', 'Metrics' (also highlighted with a red box), 'Outputs + logs', and 'Images'. Under the 'Model' tab, there's a 'Model summary' section with details like 'Algorithm name: MaxAbsScaler, XGBoostClassifier' and 'AUC weighted: 0.70930'. To the right of this, a large 'Run Metrics' dialog is open. It lists numerous performance metrics with their values. A red box highlights the 'View all other metrics' link under 'Model summary'.

Run Metrics	
Accuracy	0.92640
AUC macro	0.70930
AUC micro	0.95507
AUC weighted	0.70930
Average precision score macro	0.57672
Average precision score micro	0.94823
Average precision score weighted	0.90617
Balanced accuracy	0.50115
F1 score macro	0.48360
F1 score micro	0.92640
F1 score weighted	0.89159
Log loss	0.24162
Matthews correlation	0.024422
Norm macro recall	0.0022931

3. Select **accuracy_table, Chart** to review the various model performance curves, such as Precision-Recall, ROC, Calibration curve, and Gain & Lift curves.

The screenshot shows the 'Metrics' tab selected in the Azure Machine Learning studio. On the left, there's a list of metrics with checkboxes. One checkbox, 'accuracy_table', is checked and highlighted with a red box. Below the list, there's a 'View as:' section with 'Chart' selected (highlighted with a red box) and 'Table' as an option. Two charts are displayed side-by-side: 'Precision-Recall' and 'ROC'. Both charts show multiple curves for 'Weighted Average', 'Macro Average', 'Micro Average', and 'Ideal' (diagonal line). A red box highlights the 'accuracy_table' checkbox in the list of metrics.

Next Steps

Congratulations! You have trained and evaluated a binary classification model using automated machine learning. 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.

Lab 11 - Regressors Performance

Overview

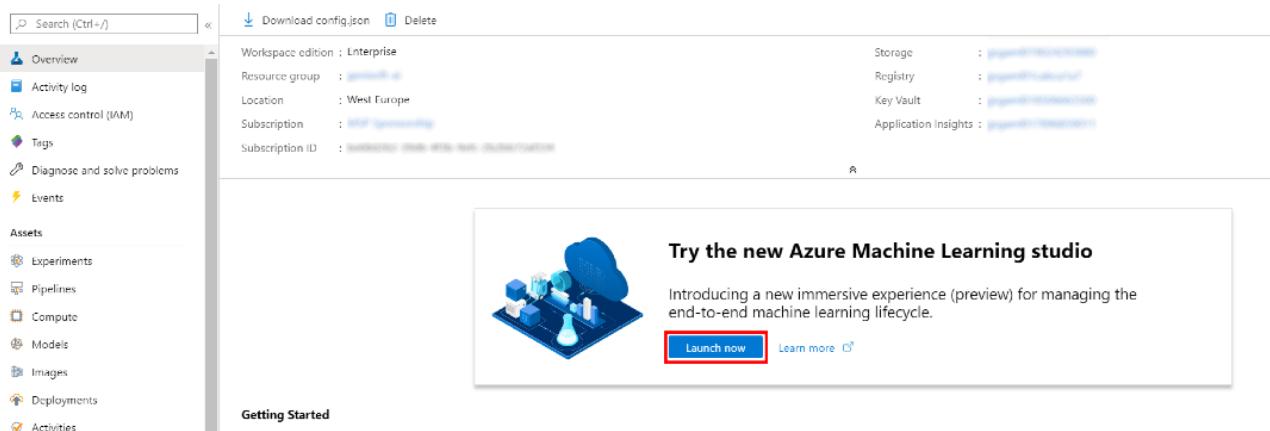
Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two regression algorithms: **Boosted Decision Tree Regression** and **Neural Net Regression** for predicting automobile prices. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Exercise 1: Create Training Pipeline

Task 1: Open Sample 2: Regression - Automobile Price Prediction (Compare algorithms)

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



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124 southcentralus
aml-quickstarts-190124

Get started

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**.

4. From the studio, select **Designer, Sample 2: Regression - Automobile Price Prediction (Compare algorithms)**.

The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, there's a sidebar with icons for creating new pipelines, managing datasets, and other tools. The main area is titled 'Designer' and shows a grid of sample pipelines. One pipeline, 'Sample 2: Regression - Automobile Price Prediction...', is highlighted with a red box. Below the grid, there's a section for 'Pipelines' with tabs for 'Pipeline drafts' and 'Pipeline runs'. A large yellow folder icon indicates 'No pipeline drafts to display'. On the right, there's a search bar and a link to 'Show more samples'.

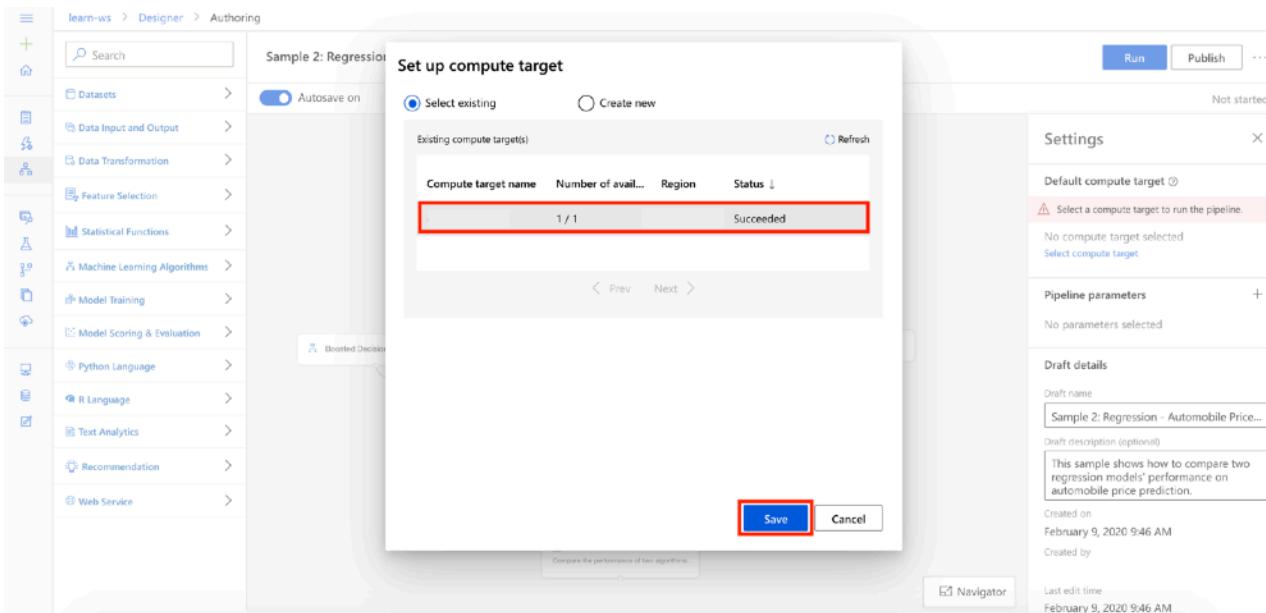
Task 2: Setup Compute Target

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

The screenshot shows the Azure Machine Learning Studio Authoring interface. On the left, there's a sidebar with various tool categories like Datasets, Data Input and Output, Feature Selection, etc. The main canvas displays a pipeline diagram for 'Sample 2: Regression - Automobile Price Prediction (Compare algorithms)'. The pipeline starts with 'Automobile price data (Raw)' and branches into two parallel paths for 'Decision Forest Regression' and 'Boosted Decision Tree Regre...'. Both paths lead to 'Score Model' and then to 'Evaluate Model'. On the right, there's a 'Settings' panel with a 'Default compute target' section that says 'Select a compute target to run the pipeline.' and 'No compute target selected'. It also includes sections for 'Pipeline parameters', 'Draft details' (with draft name 'Sample 2: Regression - Automobile Price...', draft description 'This sample shows how to compare two regression models' performance on automobile price prediction.', and creation details), and a 'Navigator' button.

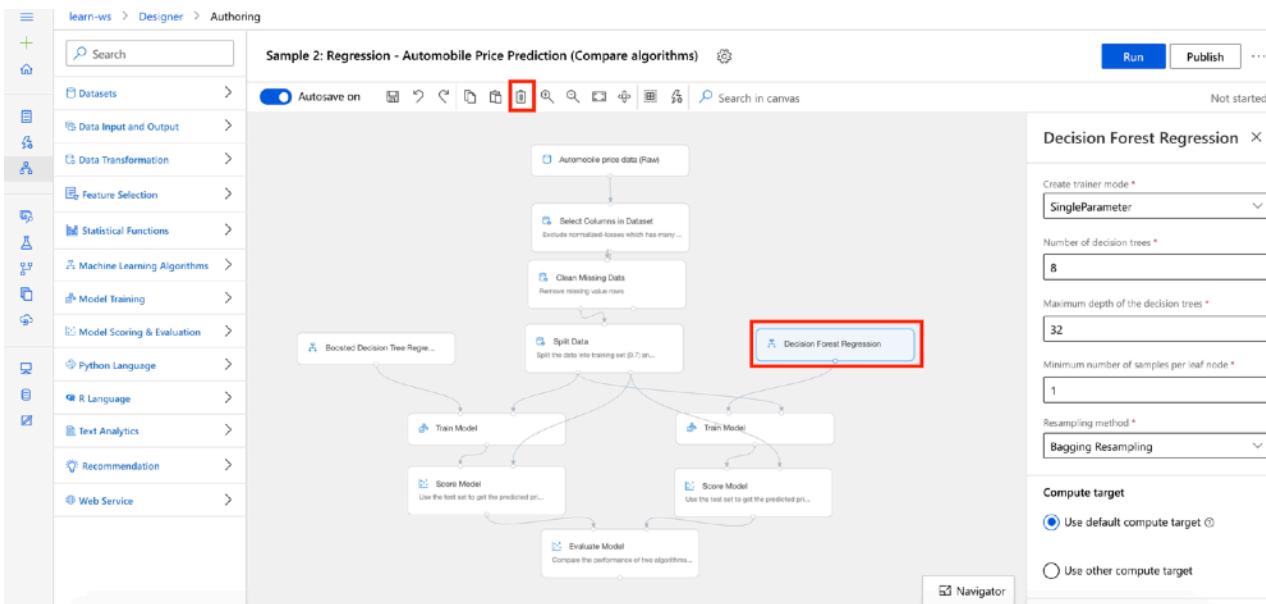
- In the **Set up compute target** editor, select the available compute, and then select **Save**.

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: Delete Pipeline Modules

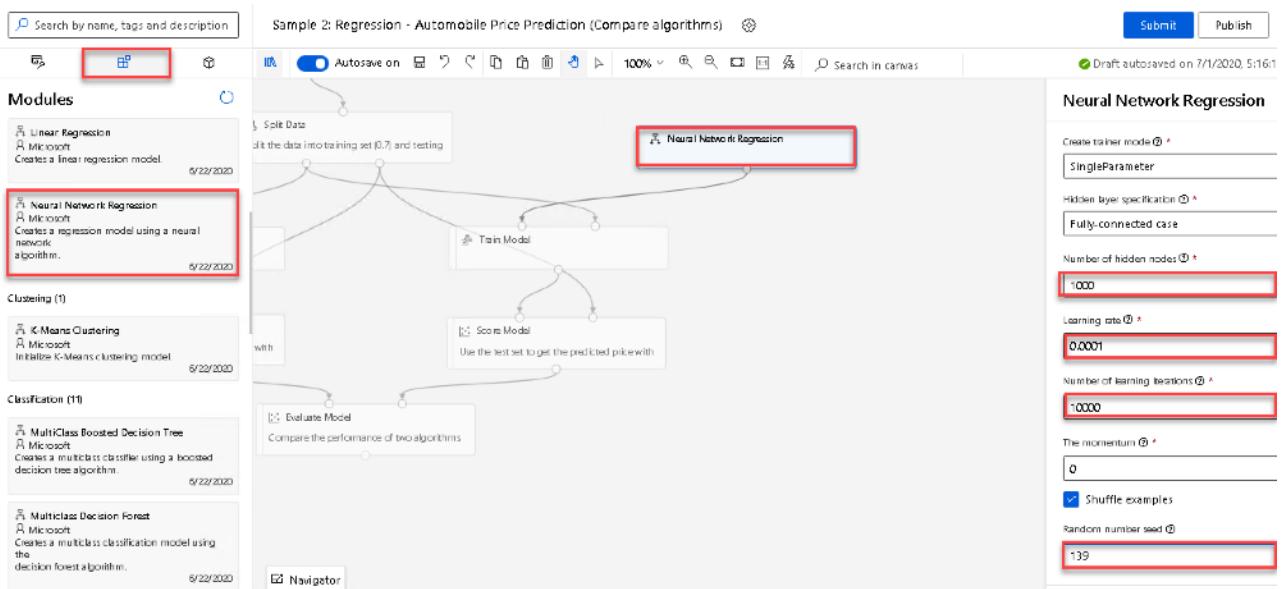
- From the **right-hand-side** of the pipeline, select the **Decision Forest Regression module** and then select the **Delete Icon**.



Task 4: Setup the Neural Net Regression Module

- Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 - Select the **Neural Net Regression** prebuilt module
 - Drag and drop the selected module on to the canvas
 - Set **Number of hidden nodes** to **1000**
 - Set **Learning rate** to **0.0001**
 - Set **Number of learning iterations** to **10000**
 - Set **Random number seed** to **139**

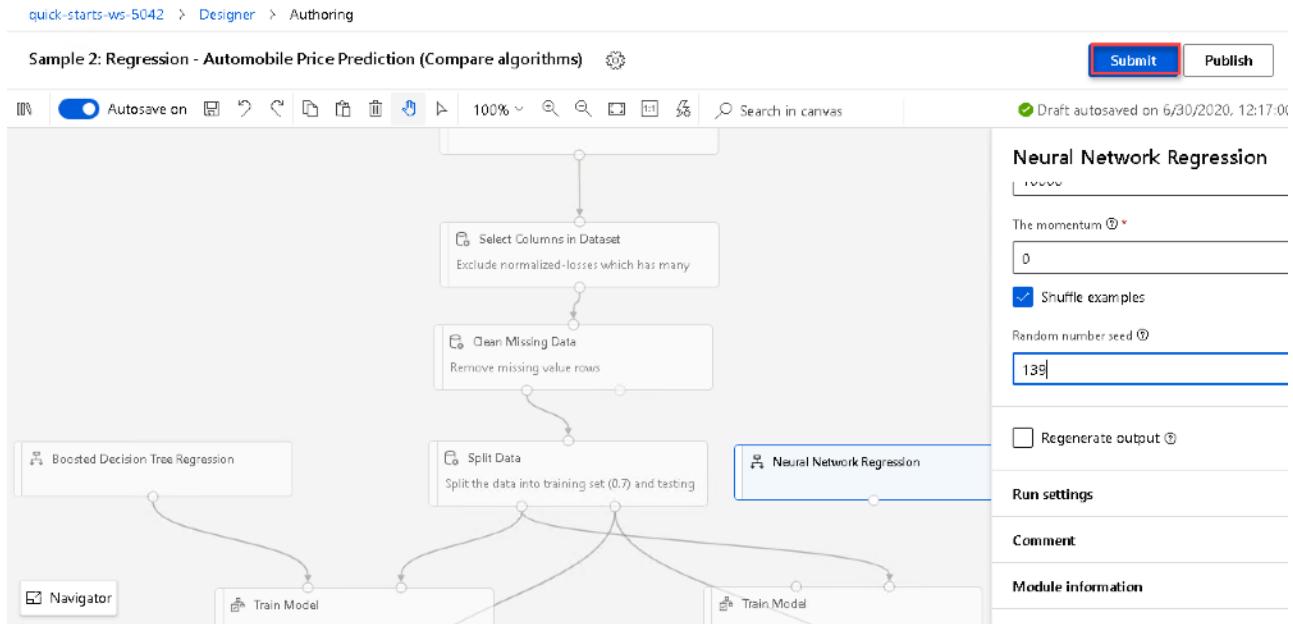
7. Connect the Neural Net Regression module to the first input of the Train Model module



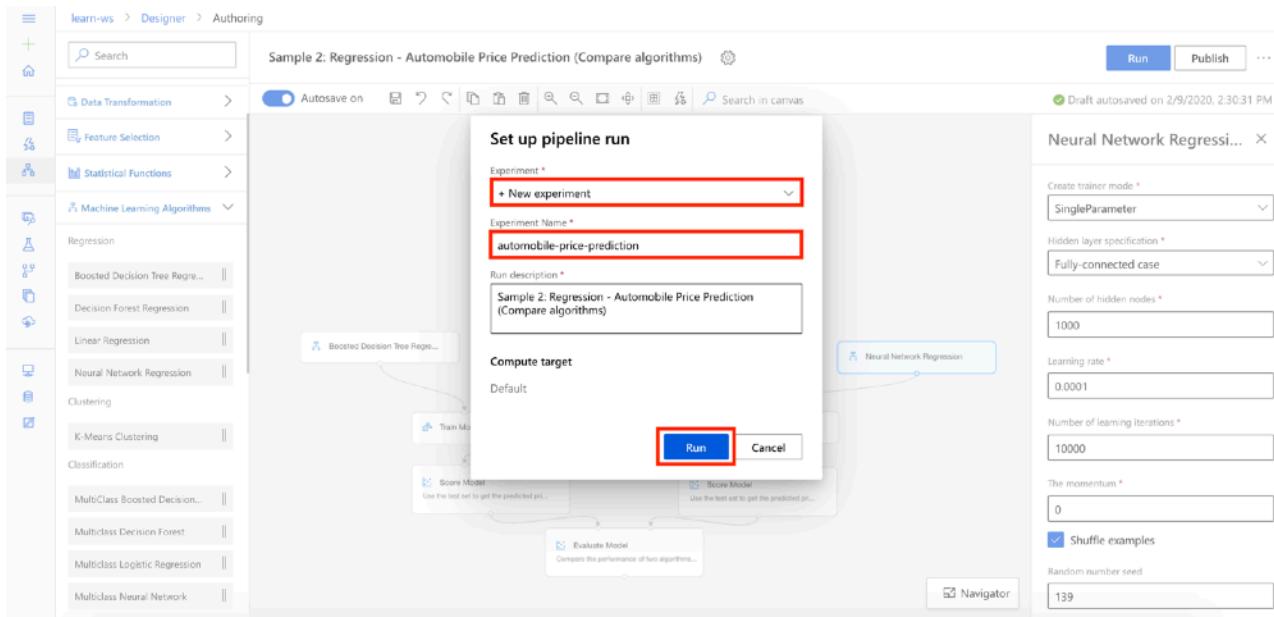
Exercise 2: Submit Training Pipeline

Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run** editor.



2. Please note that the button name in the UI is changed from **Run** to **Submit**.
3. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: automobile-price-prediction**, and then select **Submit**.

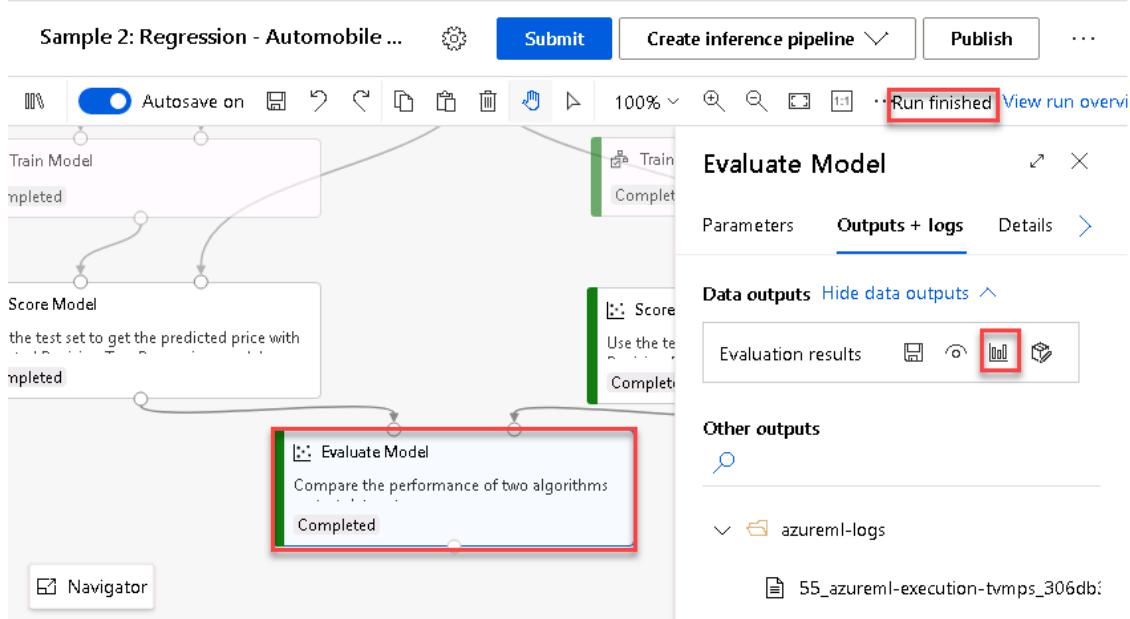


4. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.
5. While you wait for the model training to complete, you can learn more about the evaluation metrics for the regression algorithm used in this lab by selecting [Metrics for regression models](#).

Exercise 3: Compare Model Performance

Task 1: Open Evaluation Results

1. Select **Evaluate Model, Outputs, Visualize** to open the **Evaluate Model result visualization** dialog.

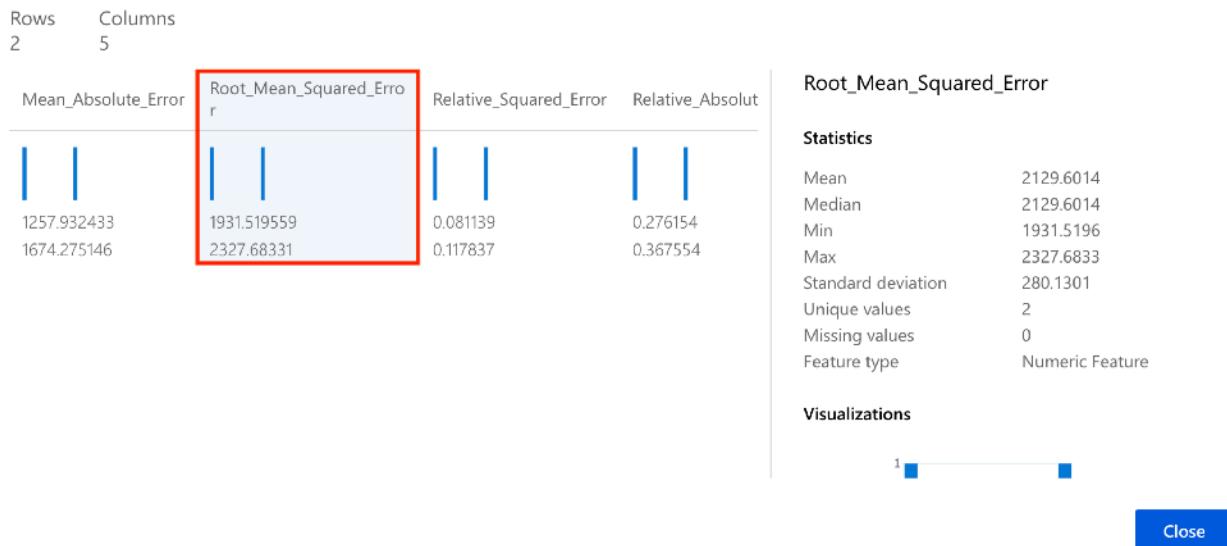


Task 2: Compare Performance Metrics

1. Select the regression performance metric **Root_Mean_Squared_Error** and compare performance of the two algorithms: **Boosted Decision Tree Regression** and **Neural Net Regression**. Note that smaller value for **Root_Mean_Squared_Error** implies better performance.

Evaluate Model result visualization

X



Task 3: Conclusion

1. Based on the performance metric, **Root_Mean_Squared_Error**, it shows that the **Boosted Decision Tree Regression** algorithm outperforms the **Neural Net Regression** algorithm. One recommendation for next steps is to tune the hyperparameters for the **Neural Net Regression** module to see if we can improve its performance.

Next Steps

Congratulations! You have trained and compared performance of two different regression machine learning models. 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.

Lab 12 - Train a Regressor using Automated Machine Learning

Overview

Automated machine learning picks an algorithm and hyperparameters for you and generates a model ready for deployment. There are several options that you can use to configure automated machine learning experiments.

Configuration options available in automated machine learning:

- Select your experiment type: Classification, Regression or Time Series Forecasting
- Data source, formats, and fetch data
- Choose your compute target
- Automated machine learning experiment settings
- Run an automated machine learning experiment
- Explore model metrics
- Register and deploy model

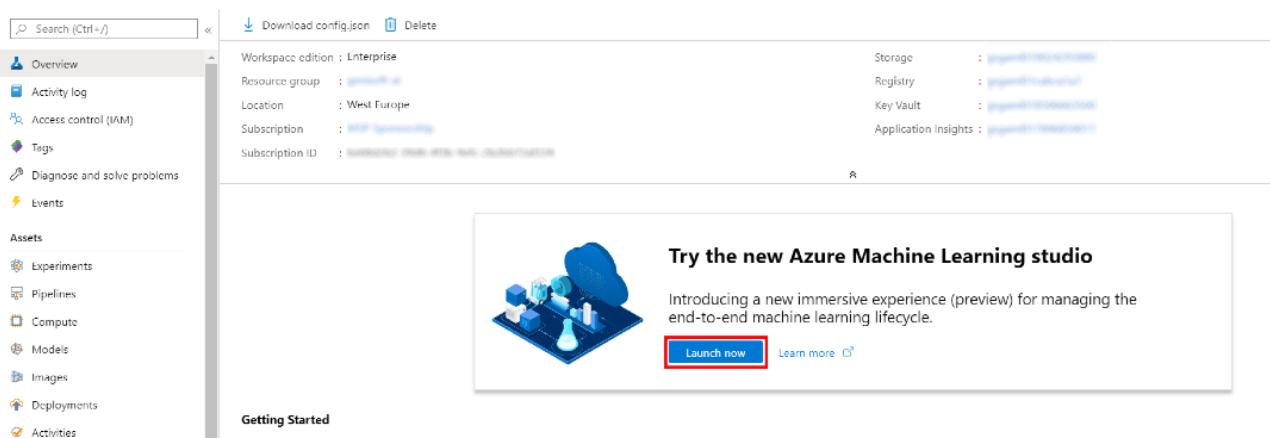
You can create and run automated machine learning experiments in code using the [Azure ML Python SDK](#) or if you prefer a no code experience, you can also create your automated machine learning experiments in [Azure Machine Learning Studio](#).

In this lab, we will use Automated Machine Learning to find the best performing regression model for predicting automobile prices. We will do all of this from the [Azure Machine Learning Studio](#) without writing a single line of code.

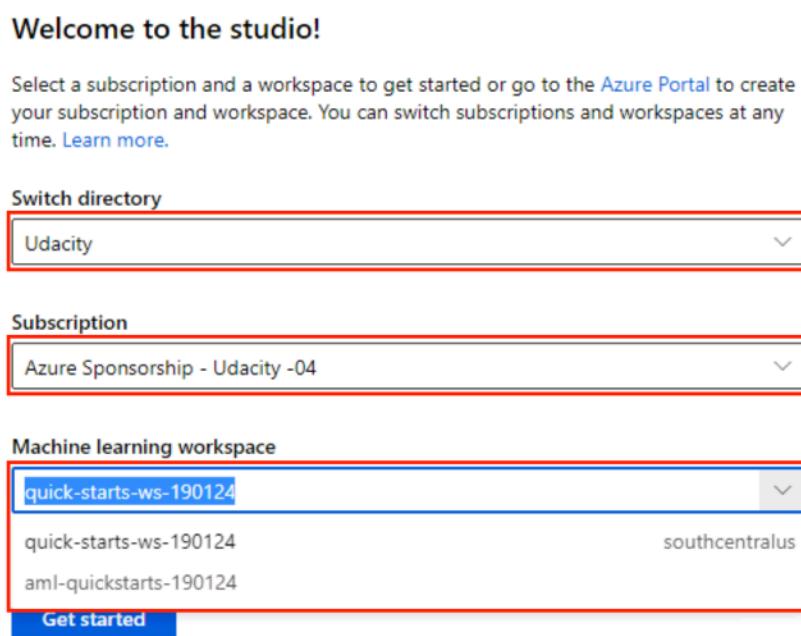
Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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



3. 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**.

4. From the studio, select **Datasets**, + **Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the Azure Machine Learning studio interface. On the left, there's a sidebar with various options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', and 'Datasets'. The 'Datasets' option is highlighted with a red box. On the right, the main area is titled 'Datasets' and shows 'Registered datasets'. There's a 'Create dataset' dropdown with four options: 'From local files', 'From datastore', 'From web files' (which is also highlighted with a red box), and 'From Open Datasets'. Below the dropdown, there's a table with columns 'Version', 'Created on', 'Modified on', 'Properties', and 'Tags'. A message at the bottom says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'.

5. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampleddata.blob.core.windows.net/data/automobile-price/automobile-price.csv>

6. Provide **Automobile-Price** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there's a sidebar with 'Datasets' and a table showing one dataset named 'CRM-Churn'. The main dialog has tabs: 'Basic info' (selected), 'Settings and preview', 'Schema', and 'Confirm details'. Under 'Basic info', there's a 'Web URL' field containing 'https://introtomlsampleddata.blob.core.windows.net/data/automobile-price/automobile-price.csv'. Below it are 'Name' (set to 'Automobile-Price'), 'Dataset type' (set to 'Tabular'), and a 'Description' field. At the bottom, there are 'Back', 'Next' (highlighted with a red box), and 'Cancel' buttons.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Review the dataset and then select **Next**

Basic info

Settings and preview

Schema

Confirm details

File format: Delimited

Delimiter: Comma Example: Field1,Field2,Field3

Encoding: UTF-8

Column headers: All files have same headers

Skip rows: None

symboling	ma...	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-locat...
3	alfa...	gas	std	two	convertible	rwd	front
3	alfa...	gas	std	two	convertible	rwd	front
1	alfa...	gas	std	two	hatchback	rwd	front
2	audi	gas	std	four	sedan	fwd	front
2	audi	gas	std	four	sedan	4wd	front

Back Next Cancel

Task 3: Select Columns

1. Keep the default selections, and select **Next**

Basic info

Settings and preview

Schema

Confirm details

Include	Column name	Properties	Type
Path	Path	Not applicable to select...	String
symboling	symboling	Not applicable to select...	Integer
make	make	Not applicable to select...	String
fuel-type	fuel-type	Not applicable to select...	String
aspiration	aspiration	Not applicable to select...	String
num-of-doors	num-of-doors	Not applicable to select...	String
body-style	body-style	Not applicable to select...	String
drive-wheels	drive-wheels	Not applicable to select...	String
engine-location	engine-location	Not applicable to select...	String
wheel-base	wheel-base	Not applicable to select...	Decimal

Back Next Cancel

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' page in the Azure Machine Learning Studio. On the left, there's a sidebar with various icons. In the center, a modal window titled 'Create dataset from web files' is open. It has a navigation bar with 'Basic info', 'Settings and preview', 'Schema', and 'Confirm details'. The 'Basic info' tab is selected, displaying the dataset name 'Automobile-Price', version '1', and a 'Web URL' pointing to a CSV file. To the right, under 'File settings', are options for file format (Delimited), delimiter (Comma), encoding (UTF-8), column headers (All files have same headers), and skip rows (None). At the bottom of the modal are 'Back', 'Create' (which is highlighted with a red box), and 'Cancel' buttons.

Exercise 2: Setup New Automated Machine Learning Experiment

Task 1: Create New Automated Machine Learning Experiment

1. From the studio home, select **Create new, Automated ML run**

The screenshot shows the 'Welcome to the studio!' page in the Azure Machine Learning studio. On the left, there's a sidebar with icons. The main area features a 'Create new' dropdown menu with options: Notebook, Pipeline, Dataset, **Automated ML run** (which is highlighted with a red box), and Data labeling project. To the right, there are four main cards: 'Notebooks' (Code with Python SDK and run sample experiments), 'Automated ML' (Automatically train and tune a model using a target metric), 'Designer' (Drag-and-drop interface from prepping data to deploying models), and 'Tutorials'. Below these are two smaller cards: 'Train your first ML model with Notebook' and 'Create, explore and deploy automated ML experiments'. At the bottom right, there's a link 'View all tutorials →'.

- This will open a **Create a new automated machine learning experiment** page

Task 2: Select Training Data

- Select the dataset **Automobile-Price** and then select **Next**

The screenshot shows the 'Create a new Automated ML run' wizard. On the left, a sidebar lists steps: 'Select dataset' (selected), 'Configure run', and 'Task type and settings'. The main area is titled 'Select dataset' with the sub-instruction 'Select a dataset from the list below, or create a new dataset. Automated ML currently only supports tabular data for authoring runs.' It includes a 'Create dataset' button and a 'Show supported datasets only' toggle. A search bar says 'Search to filter items...'. A table lists datasets: 'Automobile-Price' (selected, highlighted with a red box), Tabular, Feb 9, 2020 9:24 AM, Feb 9, 2020 9:24 AM. Navigation arrows < Prev and Next > are at the bottom. At the very bottom are 'Back', 'Next' (highlighted with a red box), and 'Cancel' buttons.

Task 3: Create a new Automated ML run

- Provide an experiment name: **automobile-price-prediction**
- Select target column: **price**
- Select compute target: **select the available compute**

The screenshot shows the 'Configure run' step of the wizard. The sidebar shows 'Configure run' (selected) and 'Task type and settings'. The main area has a sub-instruction: 'Configure the experiment. Select from existing experiments or define a new name, select the target column and the training compute to use. [Learn more on how to configure the experiment](#)'. It shows 'Dataset' set to 'Automobile-Price' (View dataset). The 'Experiment name *' field contains 'automobile-price-prediction' (highlighted with a red box). The 'Target column *' dropdown contains 'price' (highlighted with a red box). The 'Select training compute target *' dropdown is empty (highlighted with a red box). Below are buttons for 'Create a new compute' and 'Refresh compute'. At the bottom are 'Back', 'Next' (highlighted with a red box), and 'Cancel' buttons.

4. Select Next

Task 4: Setup Task type and Settings

- Select task type: **Regression**, and then select **View additional configuration settings**

The screenshot shows the 'Create a new Automated ML run' wizard. On the left, a vertical navigation bar lists steps: 'Select dataset', 'Configure run', and 'Task type and settings'. The 'Task type and settings' step is currently active. The main area is titled 'Select task type' with the sub-instruction: 'Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.' Three options are listed: 'Classification' (blue icon), 'Regression' (blue icon), and 'Time series forecasting' (blue icon). The 'Regression' option is selected and highlighted with a red box. Below each option are brief descriptions: 'To predict one of several categories in the target column, yes/no, blue, red, green.' for Classification, 'To predict continuous numeric values' for Regression, and 'To predict values based on time' for Time series forecasting. At the bottom of the main area are 'Back', 'Finish', and 'Cancel' buttons.

- This will open the **Additional configurations** dialog.
- Provide the following information and then select **Save**
 - Primary metric: **Normalized root mean squared error**
 - Exit criteria, Metric score threshold: **0.056**

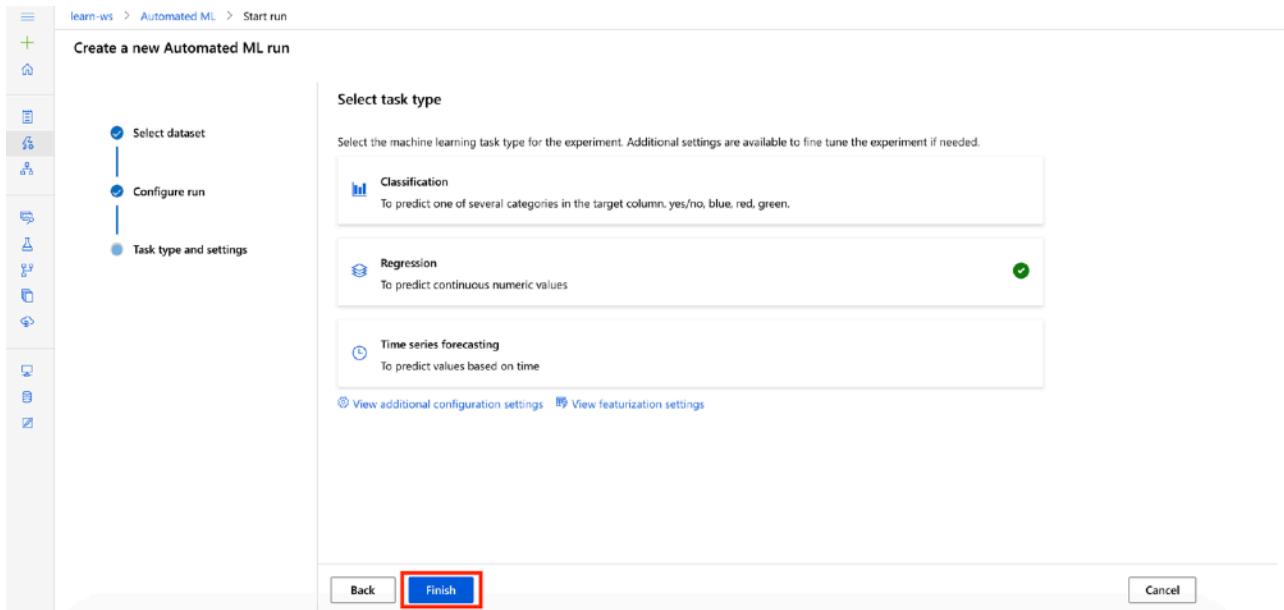
The screenshot shows the 'Create a new Automated ML run' wizard with the 'Additional configurations' dialog open. The left side of the screen shows the same navigation and task selection as the previous screenshot. The right side displays the 'Additional configurations' dialog. It includes sections for 'Primary metric' (set to 'Normalized root mean squared error'), 'Automatic featurization' (checkbox checked), 'Explain best model' (checkbox checked), 'Blocked algorithms' (a list box), 'Training job time (hours)' (set to 3), 'Metric score threshold' (set to 0.056), 'Validation', and 'Concurrency'. At the bottom right of the dialog is a large 'Save' button, which is highlighted with a red box.

- Note that we are setting a metric score threshold to limit the training time. In practice, for initial experiments, you will typically only set the training job time to allow AutoML to discover the best algorithm to use for your specific data.

Exercise 3: Start and Monitor Experiment

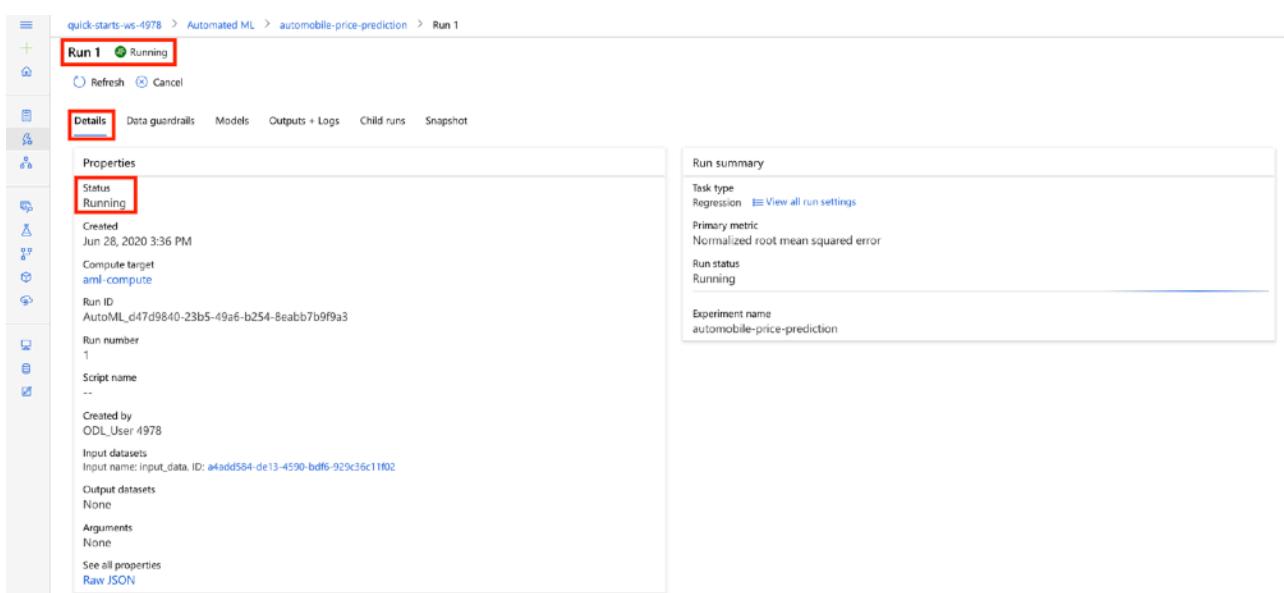
Task 1: Start Experiment

- Select **Finish** to start running the experiment



Task 2: Monitor Experiment

- The experiment will run for about 10 min.



2. In the **Details** tab, observe the **run status** of the job.
3. Select the **Models** tab, and observe the various algorithms the AutoML is evaluating. You can also observe the corresponding **Normalized root mean squared error** scores for each algorithm.

Algorithm name	Explained	Normalized root mean s...	Sampling	Run	Created	Duration	Status
MaxAbsScaler, XGBoostRegressor		0.047261	100%	Run 4	Jun 28, 2020 3:38 PM	31s	Completed
MaxAbsScaler, LightGBM		0.064467	100%	Run 3	Jun 28, 2020 3:36 PM	40s	Completed

4. Select **Details** and wait till the run status becomes **Completed**.

Run 1 Completed

Details Data guardrails Models Outputs + Logs Child runs Snapshot

Properties

- Status** Completed
- Created Jun 28, 2020 3:36 PM
- Duration 1m 52.20s
- Compute target ami-compute
- Run ID AutoML_d47d9840-23b5-49a6-b254-8eabb7b9f9a3
- Run number 1
- Script name
- Created by CDL_User 4978
- Input datasets Input name: input_data, ID: a4add584-de13-4590-bdff6-929c36c11f02
- Output datasets None
- Arguments None

Best model summary

- Algorithm name MaxAbsScaler, XGBoostRegressor
- Normalized root mean squared error 0.047261 [View all other metrics](#)
- Sampling 100% ○
- Registered models No registration yet
- Deploy status No deployment yet

Run summary

- Task type Regression [View all run settings](#)
- Primary metric Normalized root mean squared error
- Run status Completed
- Experiment name automobile-price-prediction

5. While you wait for the model training to complete, you can learn more about how Automated Machine Learning offers preprocessing and data guardrails automatically by selecting [Automatic featurization](#).

Exercise 4: Review Best Model's Performance

Task 1: Review Best Model Performance

1. From the **Details** tab review the best model's **Algorithm name** and its corresponding **Normalized root mean squared error** score. Next, select the best model's **Algorithm name**

quick-starts-ws-4978 > Automated ML > automobile-price-prediction > Run 1

Run 1 Completed

Refresh Cancel

Details Data guardrails Models Outputs + Logs Child runs Snapshot

Properties

- Status Completed
- Created Jun 28, 2020 3:36 PM
- Duration 1m 52.20s
- Compute target aml-compute
- Run ID AutoML_d47d9840-23b5-49a6-b254-8eabb7b9f9a3
- Run number 1
- Script name --
- Created by ODL_User 4978
- Input datasets Input name: input_data, ID: a4add584-de13-4590-bdf6-929c36c11f02
- Output datasets None
- Arguments None
- See all properties Raw JSON

Best model summary

- Algorithm name MaxAbsScaler, XGBoostRegressor
- Normalized root mean squared error 0.047261 [View all other metrics](#)
- Sampling 100% i
- Registered models No registration yet
- Deploy status No deployment yet

Run summary

- Task type Regression [View all run settings](#)
- Primary metric Normalized root mean squared error
- Run status Completed
- Experiment name automobile-price-prediction

2. Select **View all other metrics** to review the various **Run Metrics** to evaluate the model performance. Next, select **Metrics**

Run Metrics

Run 4 Completed

Refresh Deploy Download Explain model

Model Explanations (preview) Metrics

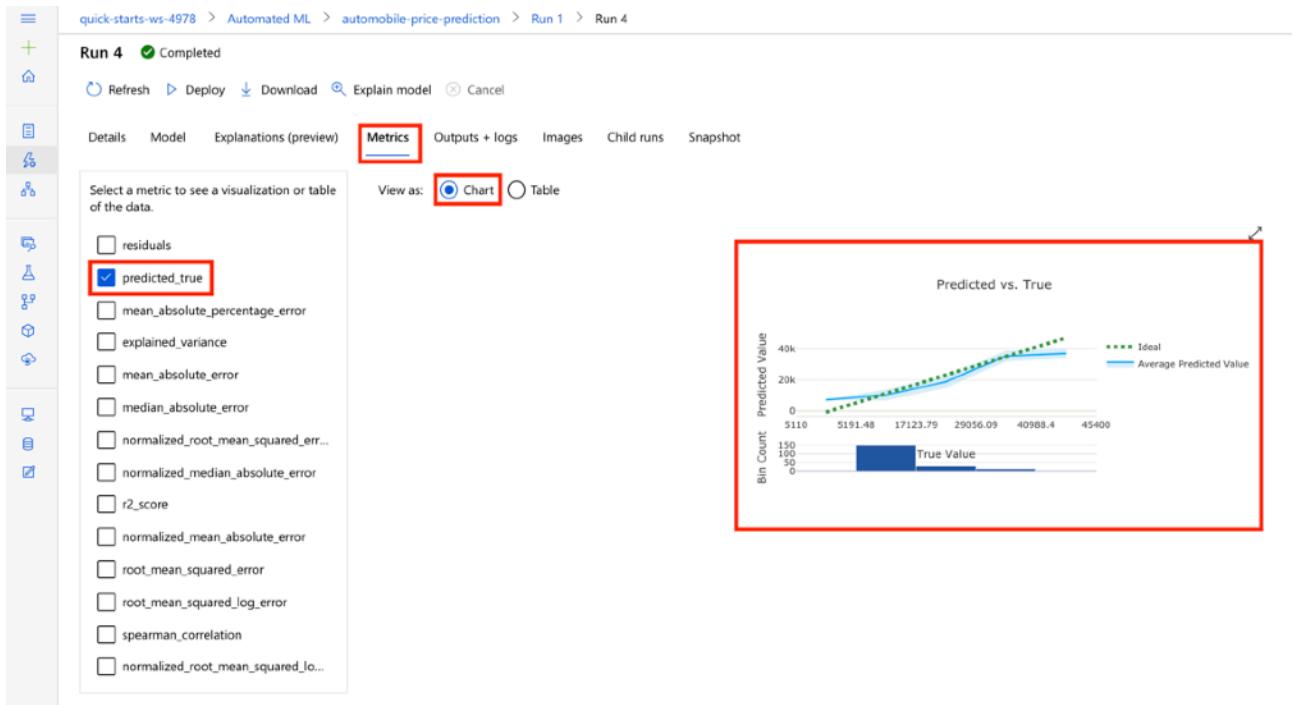
Model summary

- Algorithm name MaxAbsScaler, XGBoostRegressor
- Normalized root mean squared error 0.047261 [View all other metrics](#)
- Sampling 100% i
- Registered models No registration yet
- Deploy status No deployment yet

Run Metrics
Explained variance 0.90619
Mean absolute error 1372.2
Mean absolute percentage error 10.434
Median absolute error 952.96
Normalized mean absolute error 0.034064
Normalized median absolute error 0.023657
Normalized root mean squared error 0.047261
Normalized root mean squared log error 0.059964
R2 score 0.90338
Root mean squared error 1903.8
Root mean squared log error 0.13088
Spearman correlation 0.92256

Close

3. Select **predicted_true, Chart** to review the **Predicted vs. True** curve.



Next Steps

Congratulations! You have trained and evaluated a regression model using automated machine learning. 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.

Lab 13 - Train a Simple Clustering Model

Overview

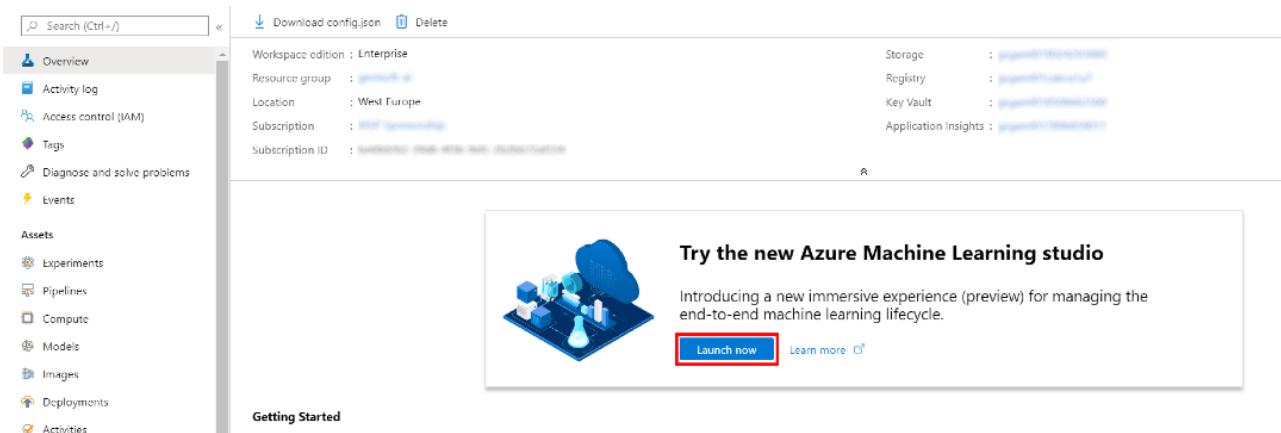
Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be using the [Weather Dataset](#) that has weather data for 66 different airports in the USA from April to October 2013. We will cluster the dataset into 5 distinct clusters based on key weather metrics, such as visibility, temperature, dew point, wind speed etc. The goal is to group airports with similar weather conditions. 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](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

Switch directory

Udacity

Subscription

Azure Sponsorship - Udacity -04

Machine learning workspace

quick-starts-ws-190124

quick-starts-ws-190124

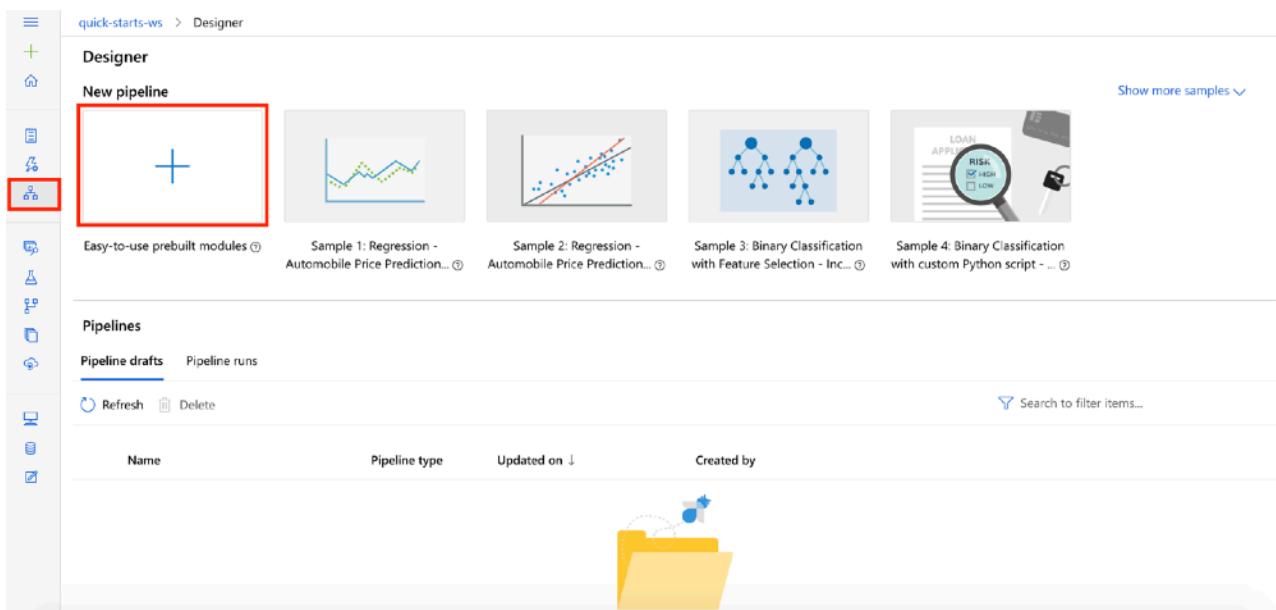
southcentralus

aml-quickstarts-190124

Get started

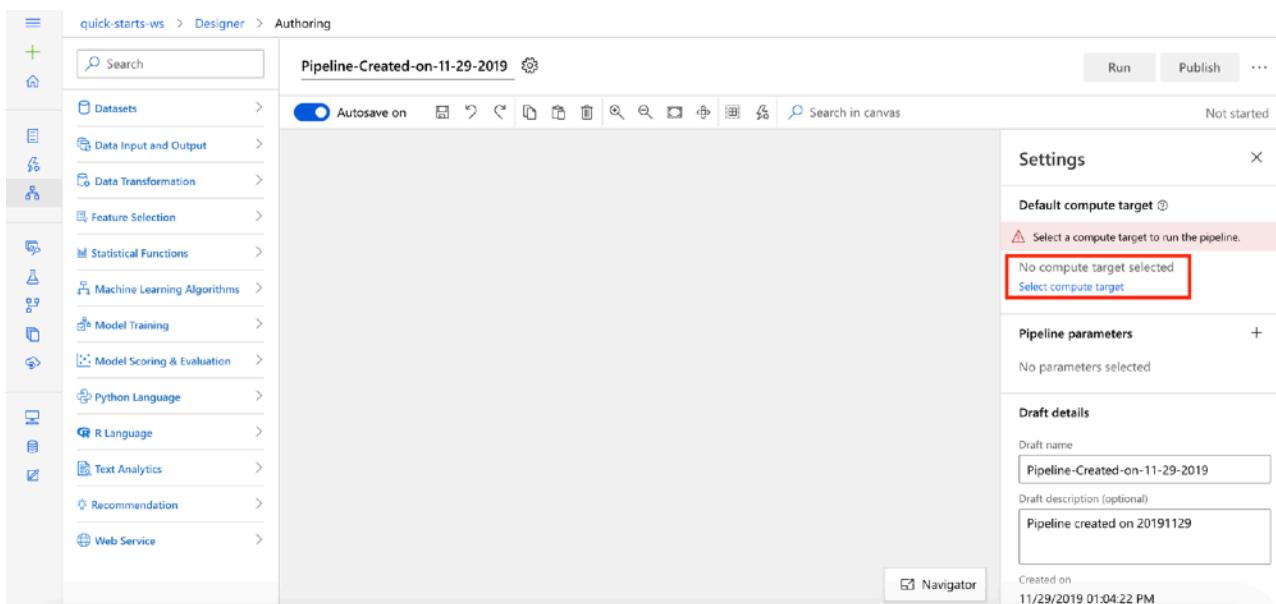
For the directory, select **Udacity** and for the subscription, select **Azure Sponsorship**. For the machine learning workspace, you may see multiple options listed. **Select any of these** (it doesn't matter which) and then click **Get started**.

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

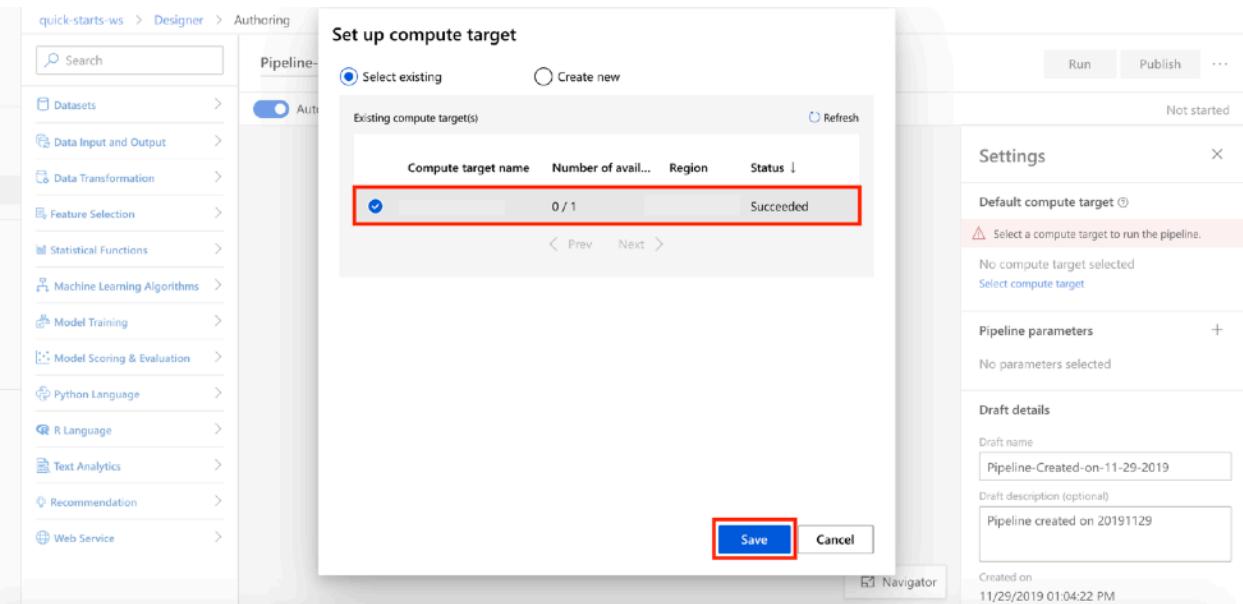


Task 2: Setup Compute Target

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



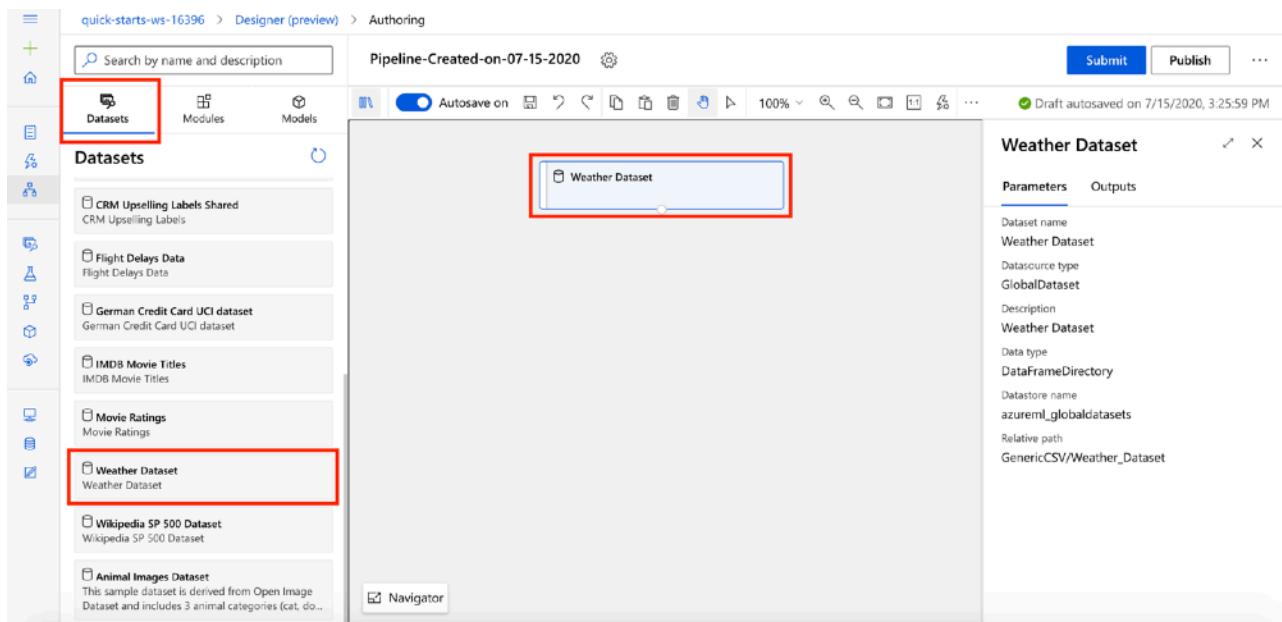
2. In the **Set up compute target** editor, select the available compute, and then select **Save**.



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 Dataset

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



Task 4: Select Columns in Dataset

1. Select **Modules, 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 **Weather Dataset** module to the **Select Columns in Dataset** module

4. Select **Edit column** link to open the **Select columns** editor

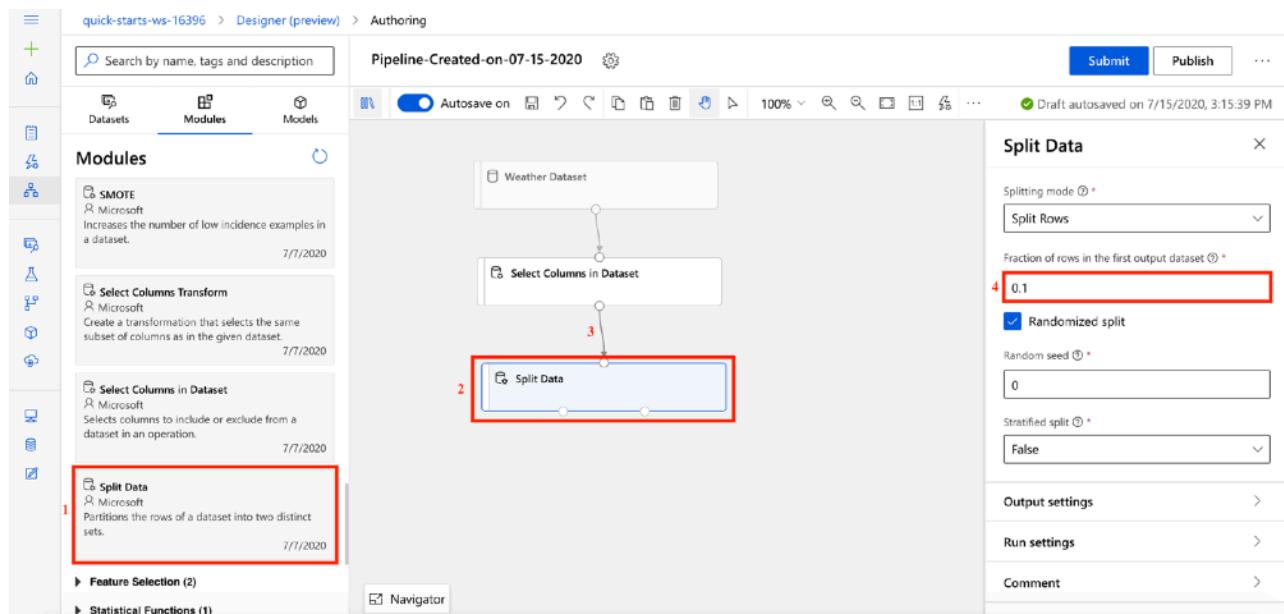
The screenshot shows the Azure Machine Learning Studio Designer interface. On the left, there's a sidebar with 'Modules' selected. In the main workspace, a pipeline is being built. A 'Select Columns in Dataset' module is highlighted with a red box and labeled '2'. Its output port is connected to a 'Weather Dataset' input port, also highlighted with a red box and labeled '3'. A tooltip 'A value is required.' is shown above the 'Edit column' link in the 'Select Columns in Dataset' module's properties panel, which is also highlighted with a red box and labeled '4'.

2. 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.
3. In the **Select columns** editor, follow the steps outlined below:
 1. Include: **Column indices**
 2. Provide column indices: **8, 10-17, 20, 26**
 3. Select **Save**

The screenshot shows the 'Select columns' editor dialog. At the top, there are two radio buttons: 'Select columns' (unchecked) and 'With rules' (checked). Below that is a toggle switch for 'Allow duplicates and preserve column order in selection'. Under the 'Include' section, a dropdown menu shows 'Column indices' and a list box contains the values '8, 10-17, 20, 26'. At the bottom right are 'Save' and 'Cancel' buttons, with 'Save' highlighted with a red box.

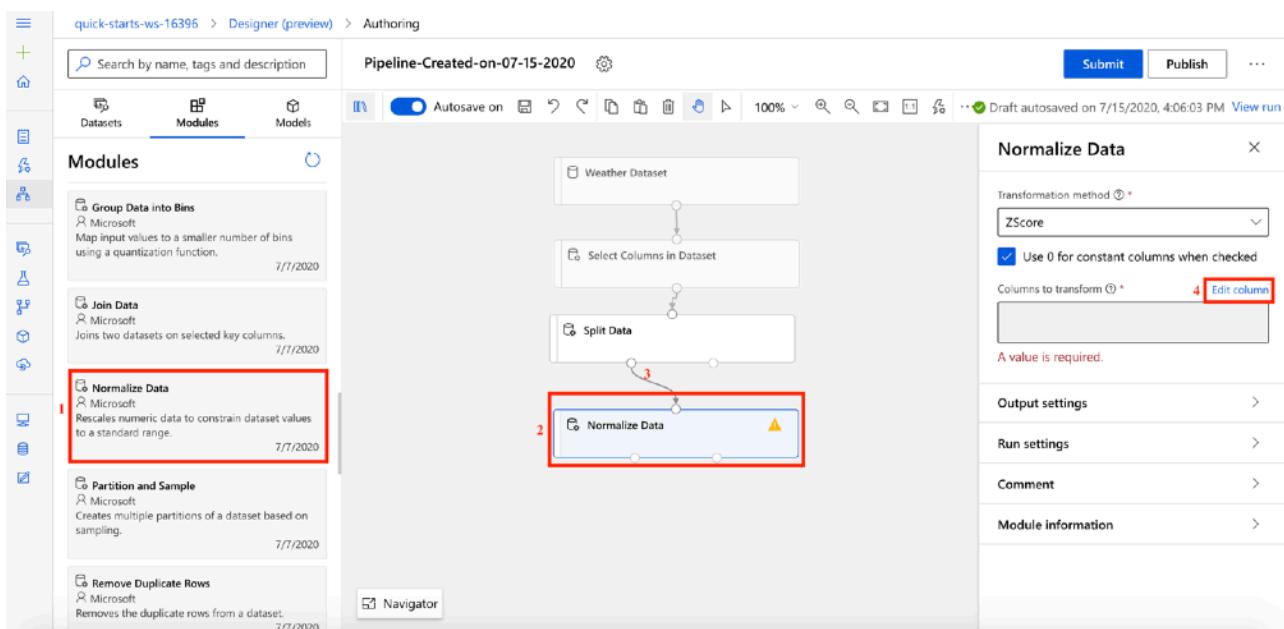
Task 5: Split Data

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. Connect the **Select Columns in Dataset** module to the **Split Data** module
 4. Fraction of rows in the first output dataset: **0.1**

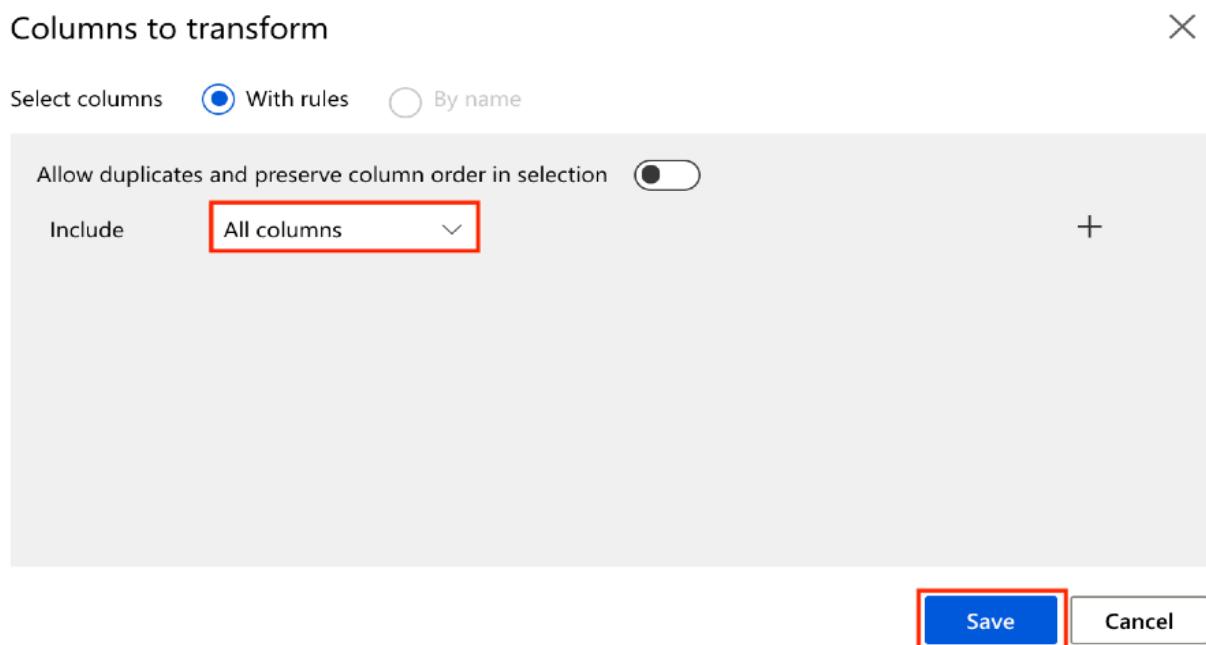


Task 6: Normalize Data

1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Normalize Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Split Data** module to the **Normalize Data** module
 4. Select **Edit column** link to open the **Columns to transform** editor

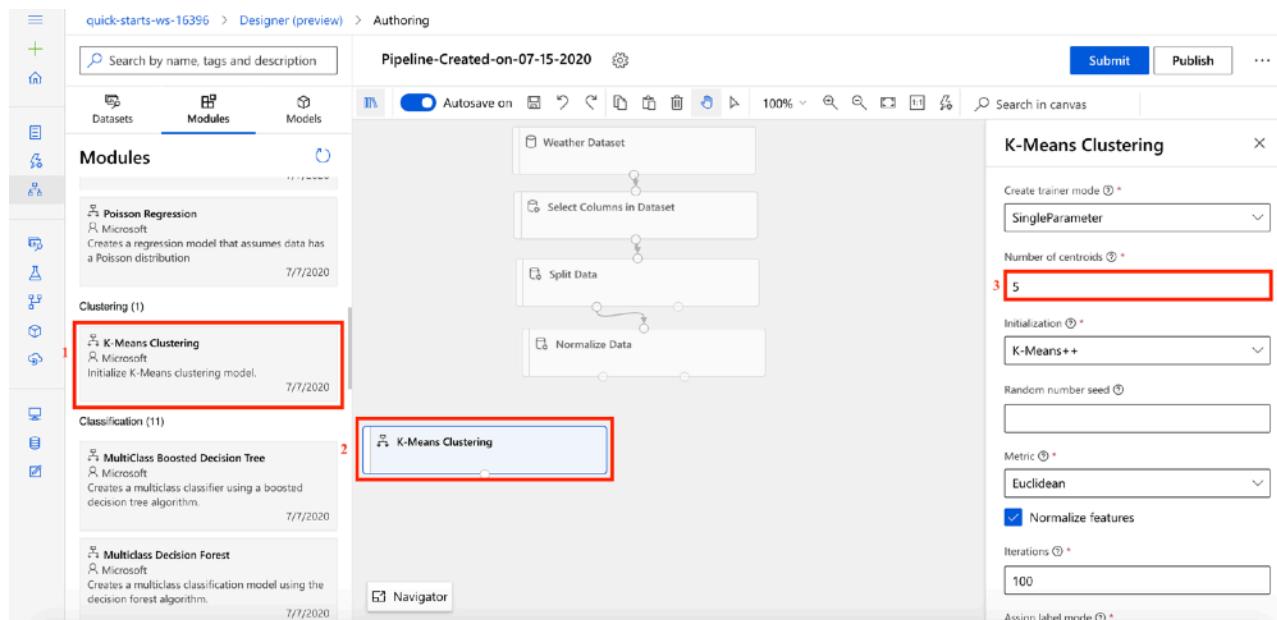


2. In the **Columns to transform** editor, follow the steps outlined below:
 1. Include: **All columns**
 2. Select **Save**



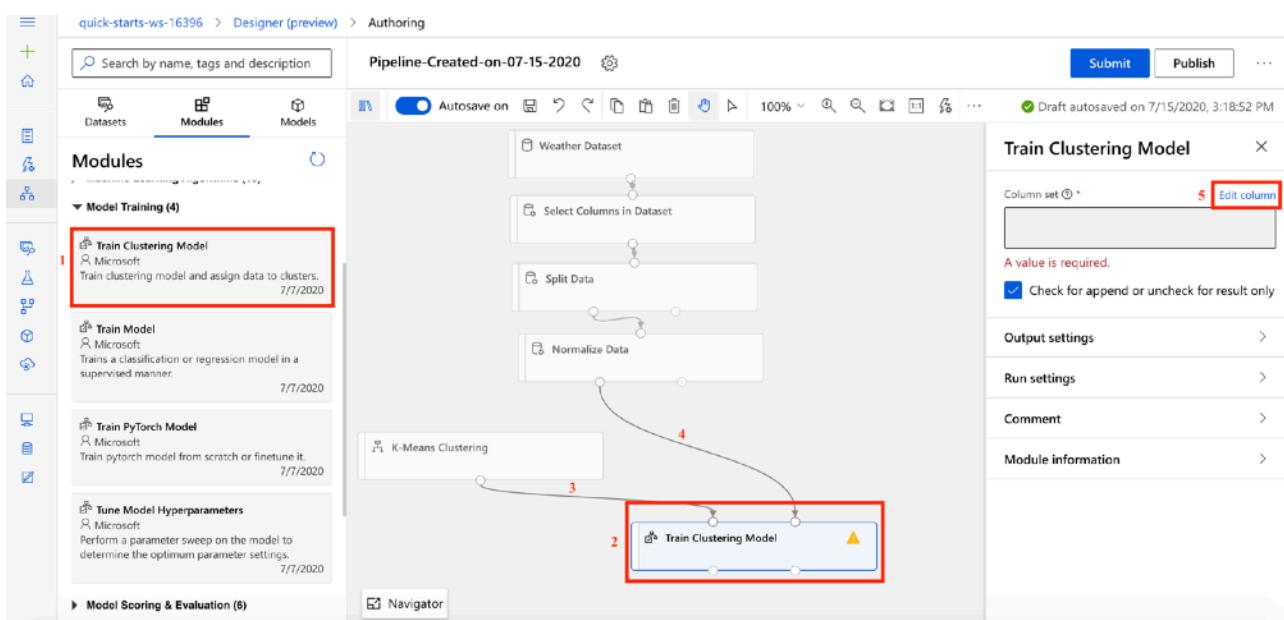
Task 7: Initialize K-Means Clustering Model

1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **K-Means Clustering** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Number of centroids: **5**



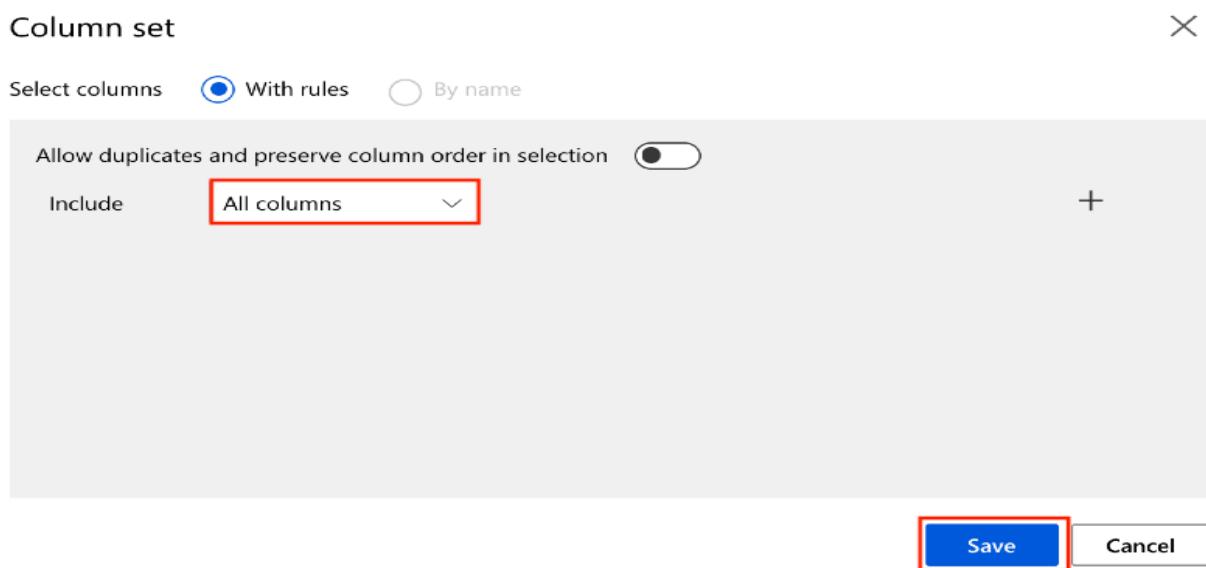
Task 8: Setup Train Clustering Model Module

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Clustering Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **K-Means Clustering** module to the first input of the **Train Clustering Model** module
 4. Connect the first output of the **Normalize Data** module to the second input of the **Train Clustering Model** module
 5. Select the **Edit column** link to open the **Column set** editor



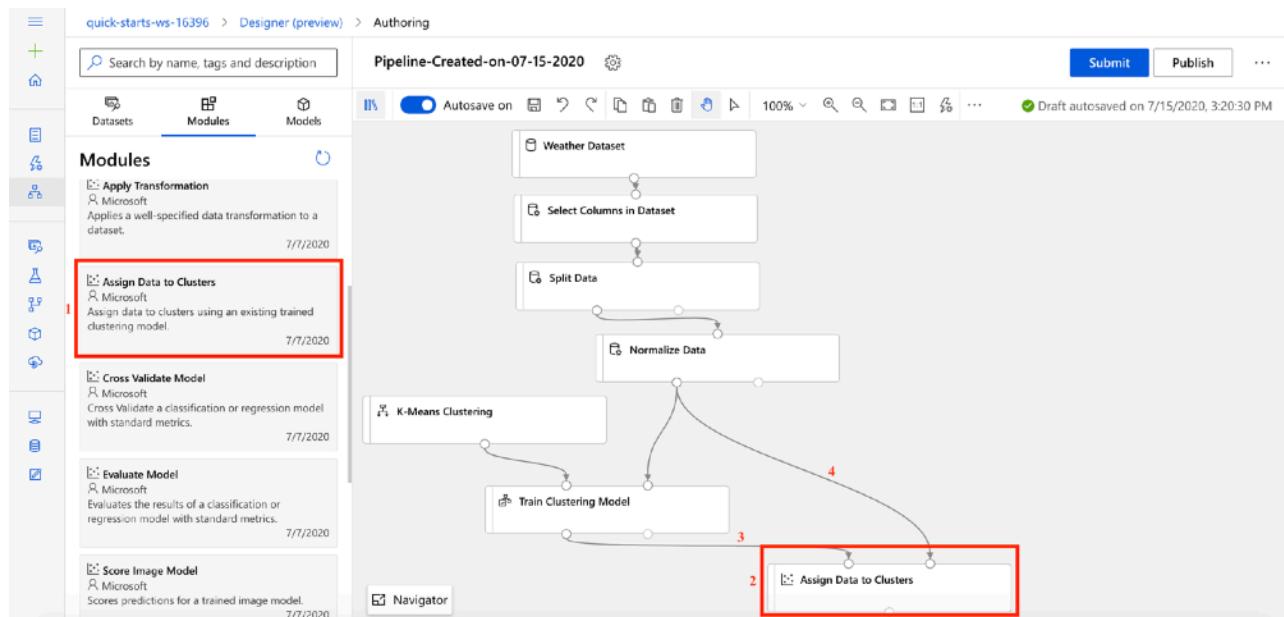
2. In the **Columns set** editor, follow the steps outlined below:

1. Include: **All columns**
2. Select **Save**



Task 9: Setup Assign Data to Clusters Module

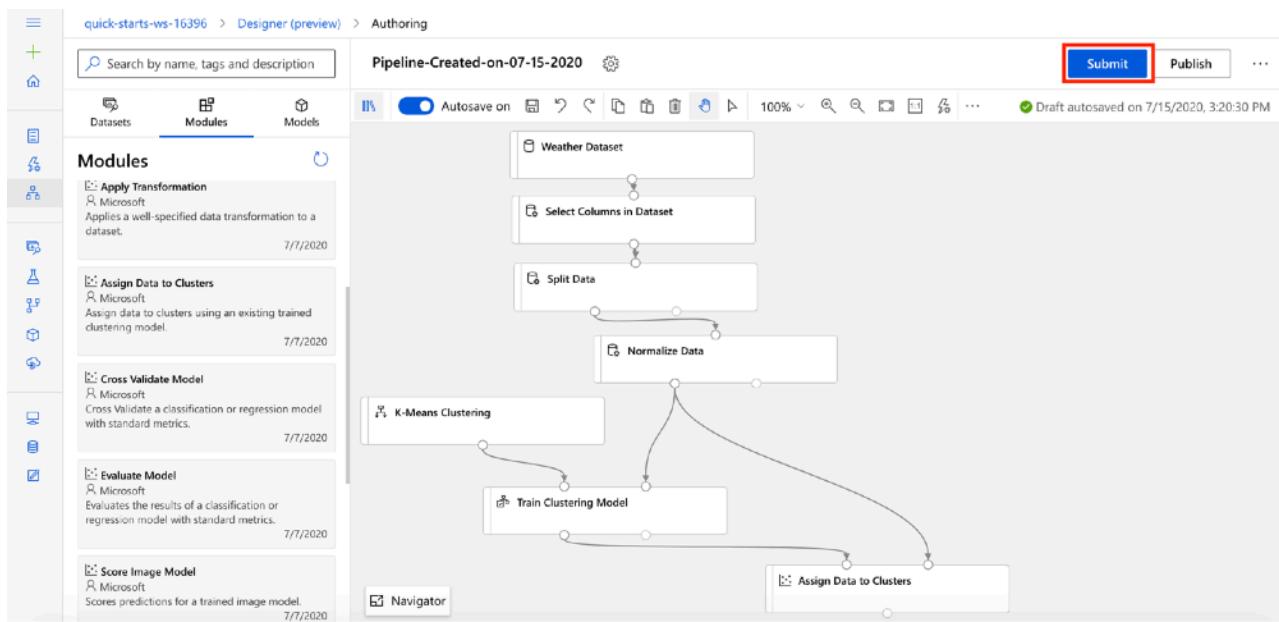
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Assign Data to Clusters** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the first output of the **Train Clustering Model** module to the first input of the **Assign Data to Clusters** module
 4. Connect the first output of the **Normalize Data** module to the second input of the **Assign Data to Clusters** module



Exercise 2: Submit Training Pipeline

Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run** editor.



2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: cluster-weather**, and then select **Submit**.

Set up pipeline run X

Experiment

Select existing Create new

New experiment name * cluster-weather

Run description * Pipeline-Created-on-07-15-2020

Compute target

Default aml-compute

Submit Cancel

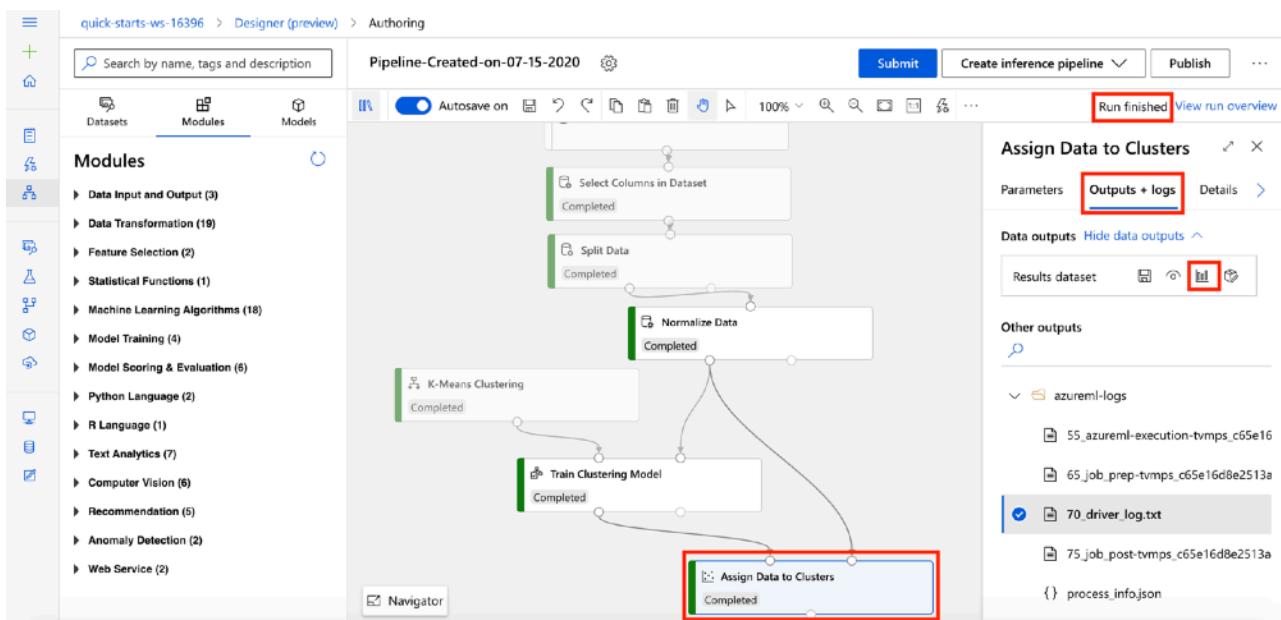
3. Wait for pipeline run to complete. It will take around **10 minutes** to complete the run.

- While you wait for the model training to complete, you can learn more about the K-Means Clustering algorithm used in this lab by selecting [K-Means Clustering](#).

Exercise 3: Visualize the Clustering Results

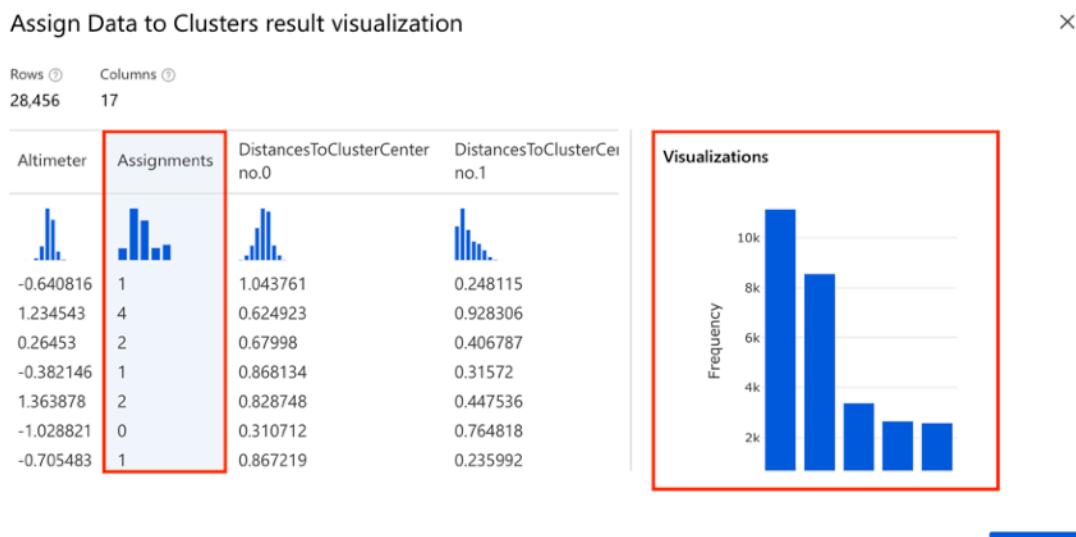
Task 1: Open the Visualization Dialog

- Select **Assign Data to Clusters, Outputs + logs, Visualize** to open the **Assign Data to Clusters result visualization** dialog.



Task 2: Evaluate Clustering Results

- Scroll to the right and select **Assignments** column.
- In the right-hand-side pane, scroll down to the **Visualizations** section.



-
3. From the results you can observe that each row (input) in the dataset is assigned to one of the 5 clusters: 0, 1, 2, 3, or 4. You can also see for each input, how far that input was from the various centroids. The cluster assignment is made based on the shortest distance between the input and cluster centroids. From the bar graph you can see the frequency distribution of all the inputs across the 5 clusters.

Next Steps

Congratulations! You have trained and evaluated your first clustering algorithm. 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.

Lab 14 - Train a simple neural net model

Although neural networks are widely known for use in deep learning and modeling complex problems such as image recognition, they are easily adapted to regression problems. Any class of statistical models can be termed a neural network if they use adaptive weights and can approximate non-linear functions of their inputs. Thus neural network regression is suited to problems where a more traditional regression model cannot fit a solution.

Neural network regression is a supervised learning method, and therefore requires a tagged dataset, which includes a label column. Because a regression model predicts a numerical value, the label column must be a numerical data type.

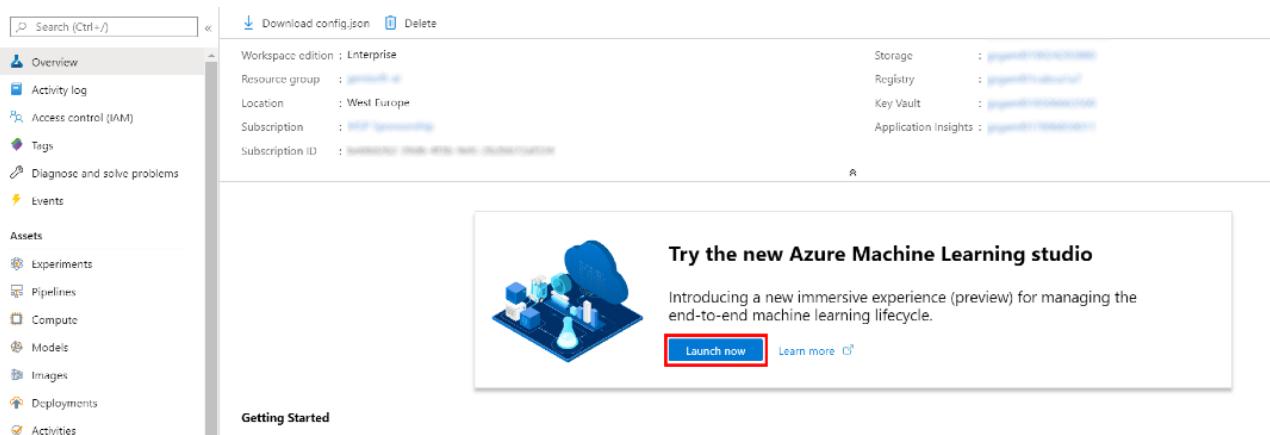
Overview

In this lab we will be using a subset of NYC Taxi & Limousine Commission - green taxi trip records available from [Azure Open Datasets](#). The data is enriched with holiday and weather data. Based on the enriched dataset, we will configure the prebuilt Neural Network Regression module to create a regression model using a customizable neural network algorithm. We will train the model by providing the model and the NYC taxi dataset as an input to Train Model. The trained model can then be used to predict NYC taxi fares. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

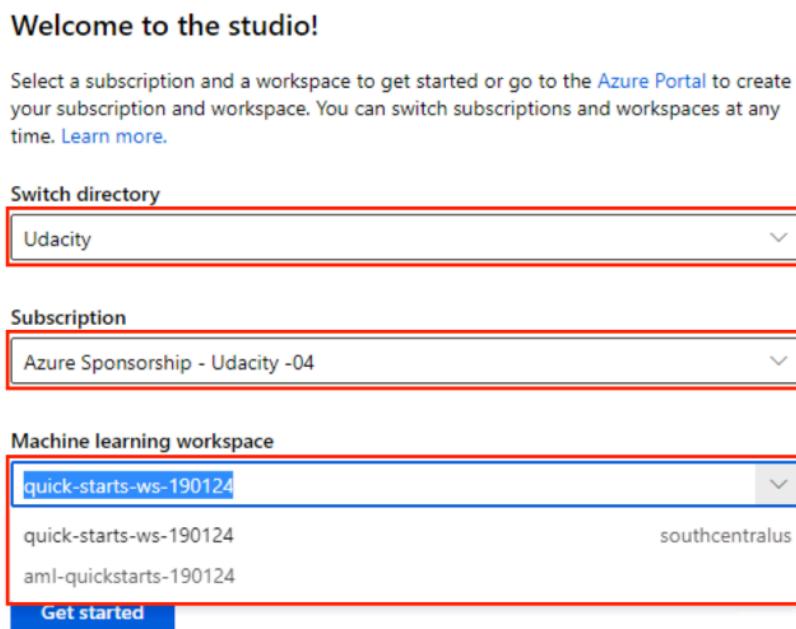
Exercise 1: Register Dataset with Azure Machine Learning studio

Task 1: Upload Dataset

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



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



4. 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**.

5. From the studio, select **Datasets**, + **Create dataset, From web files**. This will open the **Create dataset from web files** dialog on the right.

The screenshot shows the 'Datasets' page in the Azure Machine Learning Studio. On the left, there's a sidebar with various options like 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets' (which is selected and highlighted with a red box), 'Experiments', 'Pipelines', 'Models', and 'Endpoints'. The main area is titled 'Datasets' and has tabs for 'Registered datasets' and 'Dataset monitors (Preview)'. Below these are buttons for 'Create dataset' (with 'From web files' highlighted with a red box), 'Refresh', and 'Unregister'. A search bar says 'Search to filter items...'. There's a table with columns 'Version', 'Created on', 'Modified on', 'Properties', and 'Tags'. A message at the bottom says 'No datasets to display' and 'Click "Create dataset" to create your first dataset'.

6. In the Web URL field provide the following URL for the training data file:

<https://introtomlsampledata.blob.core.windows.net/data/nyc-taxi/nyc-taxi-sample-data.csv>

7. Provide **nyc-taxi-sample-data** as the Name, leave the remaining values at their defaults and select **Next**.

The screenshot shows the 'Create dataset from web files' dialog. On the left, there's a sidebar with 'Basic info' (selected and highlighted with a red box), 'Settings and preview', 'Schema', and 'Confirm details'. The main area has a 'Basic info' section with fields: 'Web URL *' containing the URL 'https://quickstartsws9073123377.blob.core.windows.net/azur.../nyc-taxi-sample-data.csv' (highlighted with a red box), 'Name *' containing 'nyc-taxi-sample-data' (highlighted with a red box), 'Dataset type *' set to 'Tabular', and a 'Description' text area containing 'Dataset description'.

Task 2: Preview Dataset

1. On the Settings and preview panel, set the column headers drop down to **All files have same headers**.
2. Scroll the data preview to right to observe the target column: **totalAmount**. After you are done reviewing the data, select **Next**

Create dataset from web files

Settings and preview

These settings were automatically detected. Please verify that the selections were made correctly or update.

File format: Delimited

Delimiter: Comma Example: Field1,Field2,Field3

Encoding: UTF-8

Column headers: All files have same headers

Skip rows: None

normalizeHolidayName	isPaidTimeOff	snowDep...	precipTime	precipDepth	temperature	totalAmount
None	false	29.06	24.00	3.00	6.19	44.30
None	false	0.00	6.00	0.00	4.57	44.80
None	false	0.00	1.00	0.00	4.38	18.96
None	false	29.06	24.00	3.00	6.19	16.30
None	false	0.00	1.00	0.00	3.85	5.30

Back Next Cancel

Task 3: Select Columns

1. Select columns from the dataset to include as part of your training data. Leave the default selections and select **Next**

quick-starts-ws > Datasets

Datasets

Registered datasets Dataset monitors

+ Create dataset Refresh Unregister

Name Version

Create dataset from local files

Basic info

Settings and preview

Schema

Confirm details

Include	Column name	Properties
Path	Path	Not applicable to select... ▾
vendorID	vendorID	Not applicable to select... ▾
passengerCount	passengerCount	Not applicable to select... ▾
tripDistance	tripDistance	Not applicable to select... ▾
hour_of_day	hour_of_day	Not applicable to select... ▾
day_of_week	day_of_week	Not applicable to select... ▾
day_of_month	day_of_month	Not applicable to select... ▾
month_num	month_num	Not applicable to select... ▾
normalizeHolidayName	normalizeHolidayName	Not applicable to select... ▾

Back Next Cancel

Task 4: Create Dataset

1. Confirm the dataset details and select **Create**

The screenshot shows the 'Datasets' blade in the Azure Machine Learning studio. On the left, there's a sidebar with various icons for creating new assets like notebooks, experiments, pipelines, and datasets. The 'Datasets' section is active, showing a table of registered datasets. On the right, a modal window titled 'Create dataset from local files' is open, showing a step-by-step process: 'Basic info', 'Settings and preview', 'Schema', and 'Confirm details'. The 'Confirm details' step is currently selected. Inside this step, there are two main sections: 'Basic info' (which lists the selected file 'nyc-taxi-sample-data.csv') and 'File settings' (which includes options for file format, delimiter, encoding, and column headers). At the bottom of the modal are 'Back', 'Create' (which is highlighted with a red box), and 'Cancel' buttons.

Exercise 2: Create New Training Pipeline

Task 1: Open Pipeline Authoring Editor

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

The screenshot shows the 'Designer' blade in the Azure Machine Learning studio. The sidebar on the left has a 'Designer' tab highlighted with a red box. Below the sidebar, there are sections for 'New pipeline' (with a plus sign icon highlighted by a red box), 'Pipelines' (showing a table of existing pipelines), and 'Samples' (listing five sample pipelines: 'Easy-to-use prebuilt modules', 'Sample 1: Regression - Automobile Price Prediction...', 'Sample 2: Regression - Automobile Price Prediction...', 'Sample 3: Binary Classification with Feature Selection - Inc...', 'Sample 4: Binary Classification with custom Python script - ...', and 'Sample 5: Binary Classification - Customer Relationship Predi...'). At the bottom right, there's a search bar with the placeholder 'Search to filter items...'.

Task 2: Setup Compute Target

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

Search Pipeline-Created-on-02-12-2020 Run Publish ...

Datasets > Data Input and Output > Data Transformation > Feature Selection > Statistical Functions > Machine Learning Algorithms > Model Training > Model Scoring & Evaluation > Python Language > R Language > Text Analytics > Recommendation > Web Service >

Autosave on Search in canvas

Settings

Default compute target ⓘ Select a compute target to run the pipeline. No compute target selected Select compute target

Pipeline parameters + No parameters selected

Draft details

Draft name Pipeline-Created-on-02-12-2020 Draft description (optional) Pipeline created on 20200212

Created on February 12, 2020 3:11 PM Created by Roxana Cojocari Last edit time February 12, 2020 3:11 PM

2. 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.

Search Pipeline-Created-on-02-12-2020 Run Publish ...

Datasets > Data Input and Output > Data Transformation > Feature Selection > Statistical Functions > Machine Learning Algorithms > Model Training > Model Scoring & Evaluation > Python Language > R Language > Text Analytics > Recommendation > Web Service >

Autosave on Search in canvas

Settings

Default compute target ⓘ Select a compute target to run the pipeline. No compute target selected Select compute target

Pipeline parameters + No parameters selected

Draft details

Draft name Pipeline-Created-on-02-12-2020 Draft description (optional) Pipeline created on 20200212

Created on February 12, 2020 3:11 PM

Set up compute target

Select existing Create new

Compute target name	Number of avail...	Region	Status
aml-compute-01	0 / 1	westeurope	Succeeded

Prev Next < >

Save Cancel

Task 3: Add Dataset

1. Select **Datasets** section in the left navigation. Next, select **My Datasets, nyc-taxi-sample-data** and drag and drop the selected dataset on to the canvas.

The screenshot shows the Azure Machine Learning Studio interface. On the left, there's a navigation bar with 'Search', 'Datasets' (which is highlighted with a red box), 'Data Input and Output', and 'Data Transformation'. Below these are various data source and transformation modules. In the center, there's a canvas with a single dataset node labeled 'nyc-taxi-sample-data' (also highlighted with a red box). On the right, there's a detailed view of the selected dataset, showing its ID, name, type, and other metadata. At the top right, there are 'Run', 'Publish', and other pipeline management buttons.

Task 4: Split Dataset

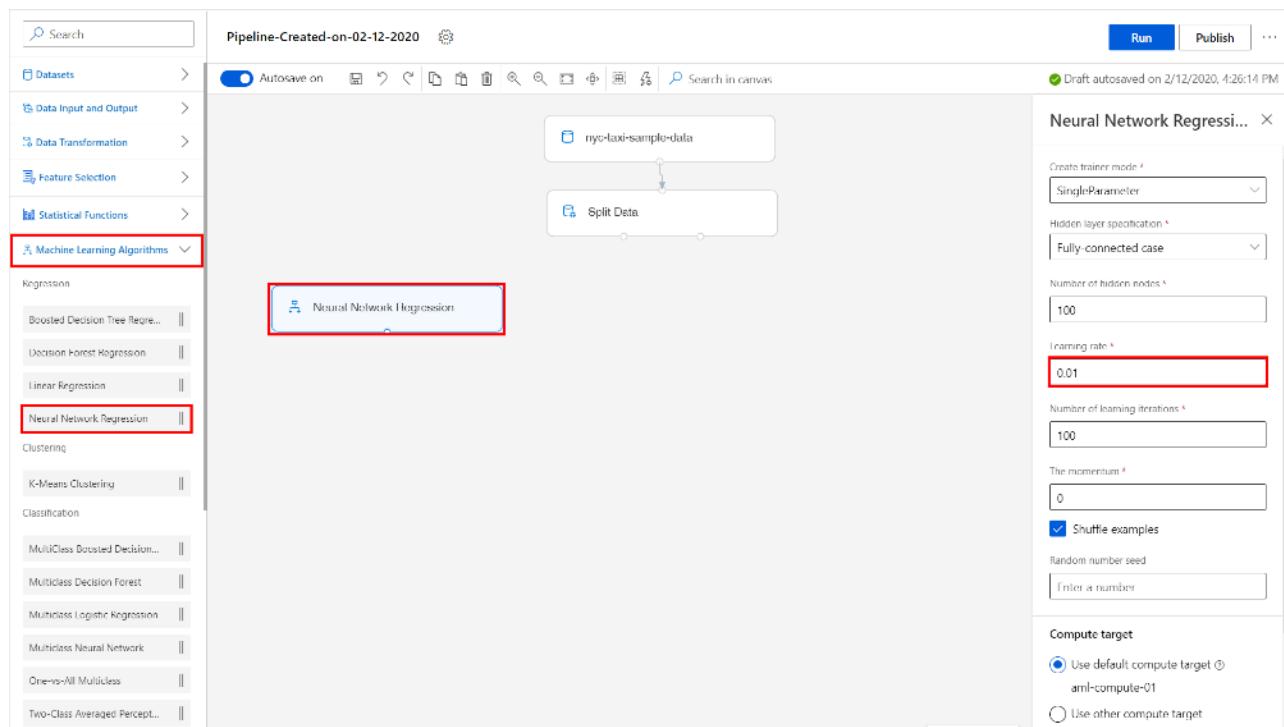
1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Split Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Fraction of rows in the first output dataset: **0.7**
 4. Connect the **Dataset** to the **Split Data** module

The screenshot shows the Azure Machine Learning Studio interface with the 'Data Transformation' section selected in the left navigation (highlighted with a red box). On the canvas, there's a 'Split Data' module (also highlighted with a red box) connected to the 'nyc-taxi-sample-data' dataset node. To the right of the canvas, there's a detailed configuration pane for the 'Split Data' module. It shows the 'Split Rows' splitting mode, a fraction of '0.7' for the first output dataset, and other options like 'Randomized split' and 'Stratified split'. Below the configuration pane, there are sections for 'Compute target' (set to 'Use default compute target: aml-compute-01') and 'Comment' (with a placeholder for a description).

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: Initialize Regression Model

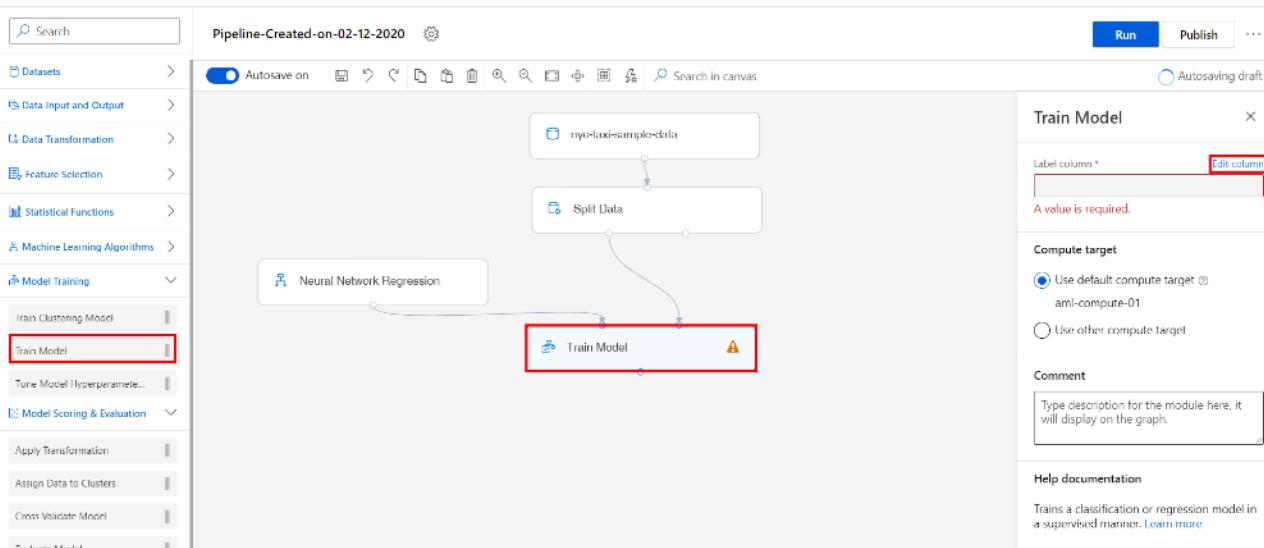
1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Neural Network Regression** prebuilt module, in the Regression category.
 2. Drag and drop the selected module on to the canvas
 3. Create trainer mode: **Single Parameter**. This option indicates how you want the model to be trained.
 4. Hidden layer specification: **Fully connected case**.
 5. For Learning rate: **0.01**.



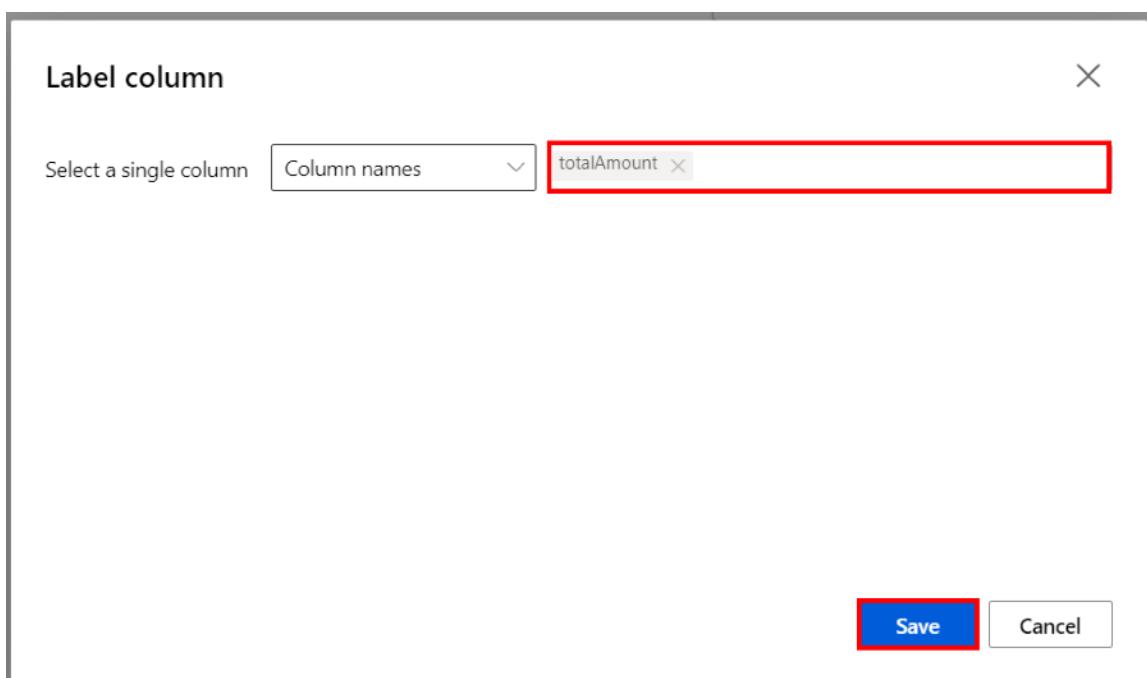
Note: Because the number of nodes in the input layer is determined by the number of features in the training data, in a regression model there can be only one node in the output layer.

Task 6: Setup Train Model Module

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Neural Network Regression** module to the first input of the **Train Model** module
 4. Connect the first output of the **Split Data** module to the second input of the **Train Model** module
 5. Select the **Edit column** link to open the **Label column** editor

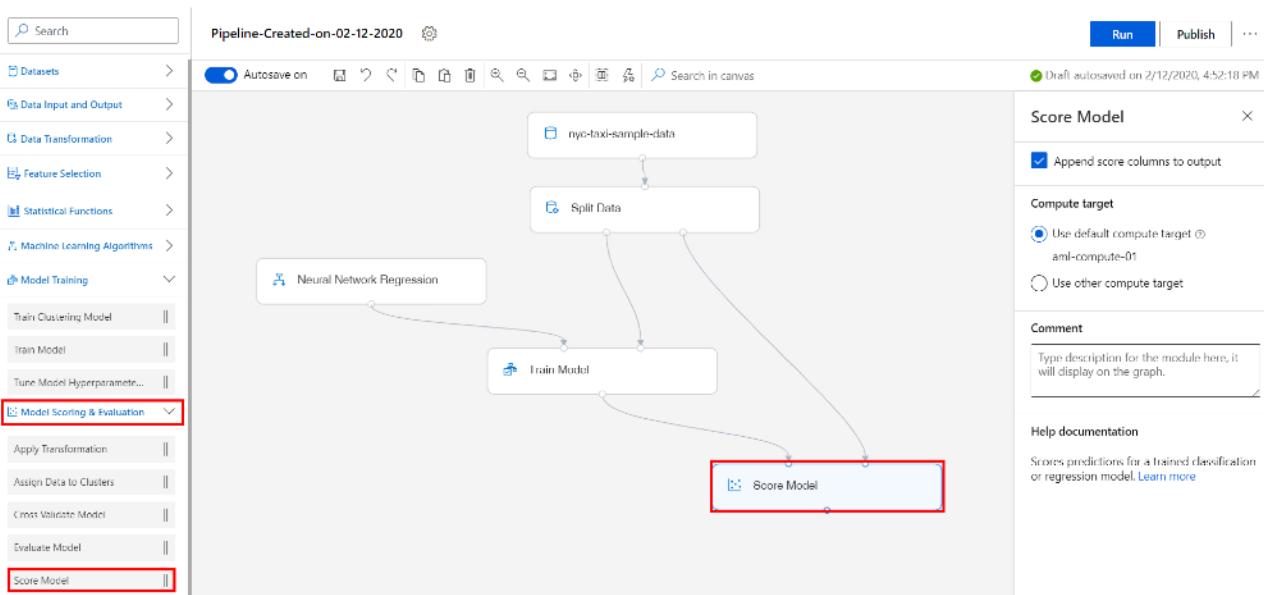


6. The **Label column** editor allows you to specify your **Label or Target column**. Type in the label column name **totalAmount** and then select **Save**.



Task 7: Setup Score Model Module

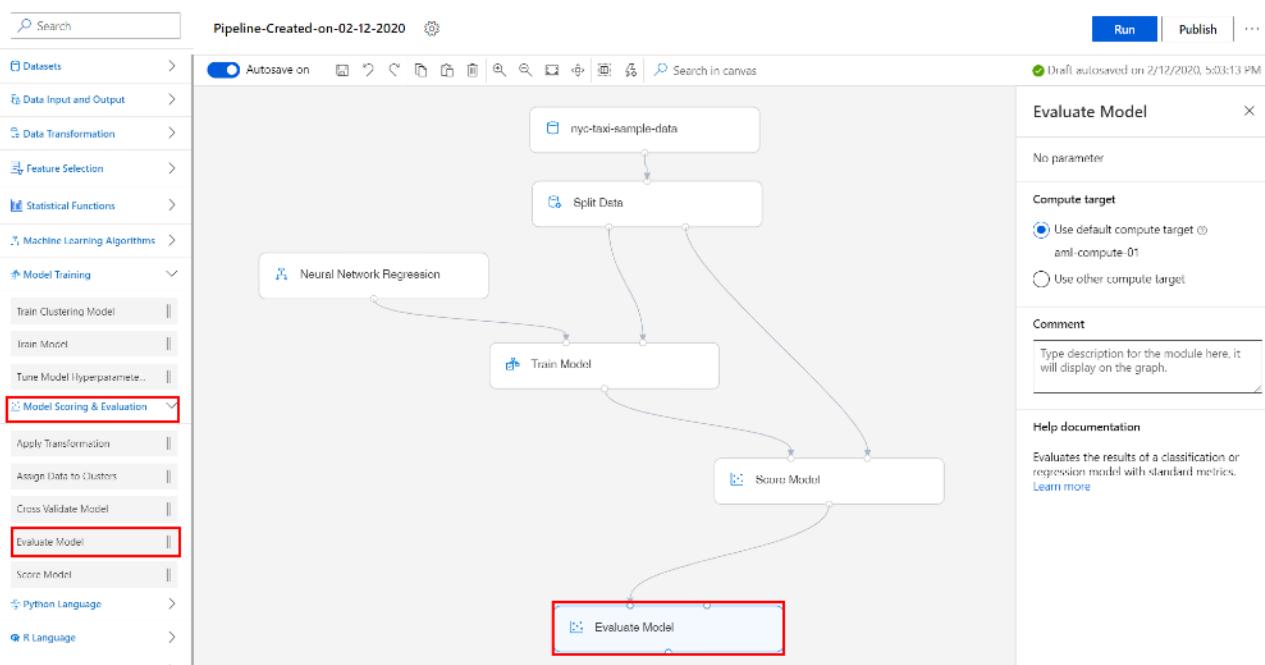
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Train Model** module to the first input of the **Score Model** module
 4. Connect the second output of the **Split Data** module to the second input of the **Score Model** module



Note that **Split Data** module will feed data for both model training and model scoring. The first output (0.7 fraction) will connect with the **Train Model** module and the second output (0.3 fraction) will connect with the **Score Model** module.

Task 8: Setup Evaluate Model Module

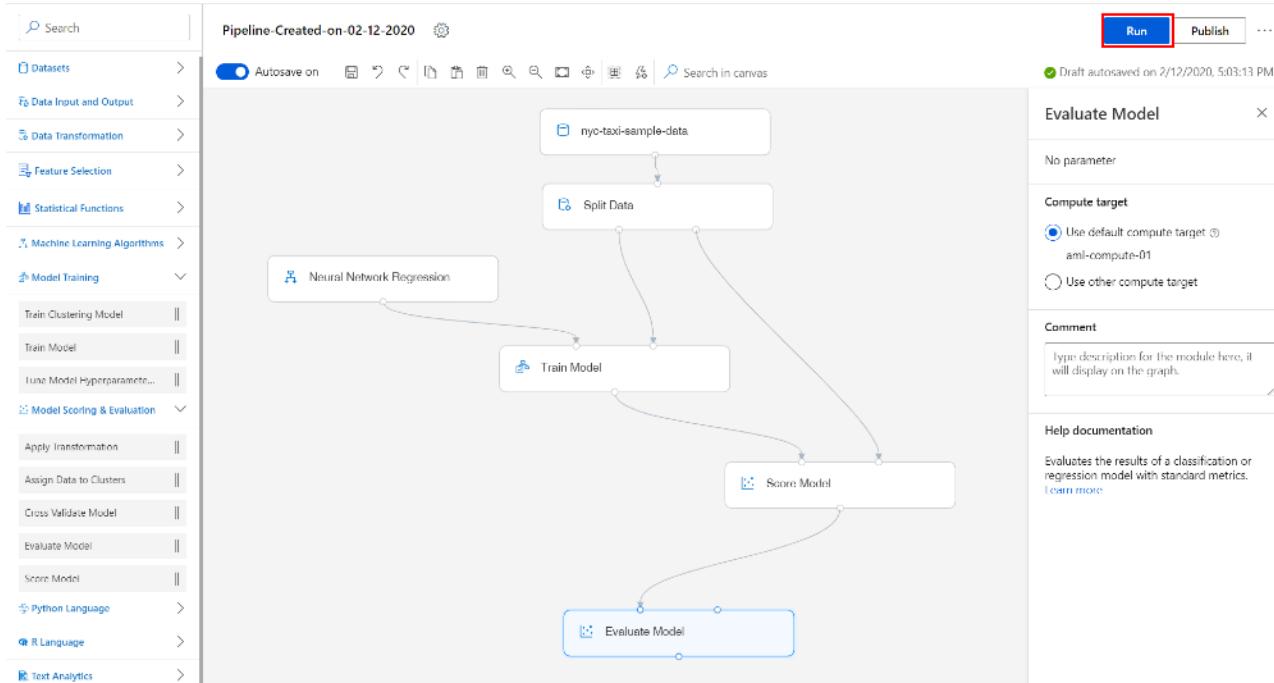
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Score Model** module to the first input of the **Evaluate Model** module



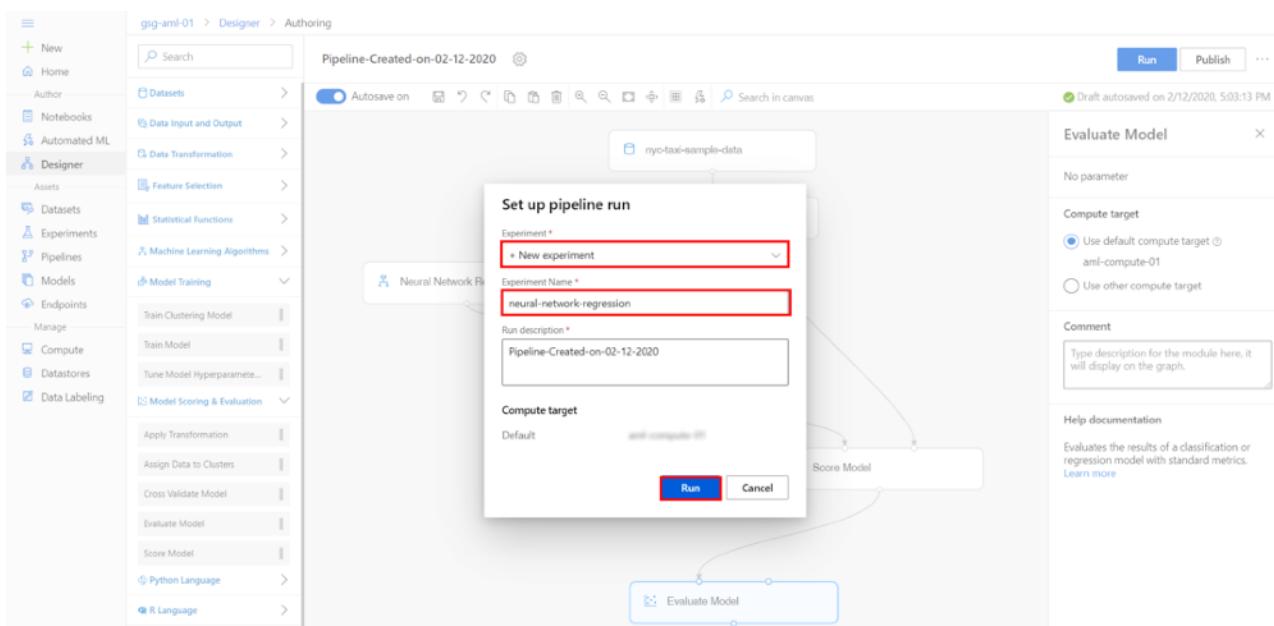
Exercise 3: Submit Training Pipeline

Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run** editor.



2. Please note that the button name in the UI is changed from **Run** to **Submit**.
3. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: neural-network-regression**, and then select **Submit**.

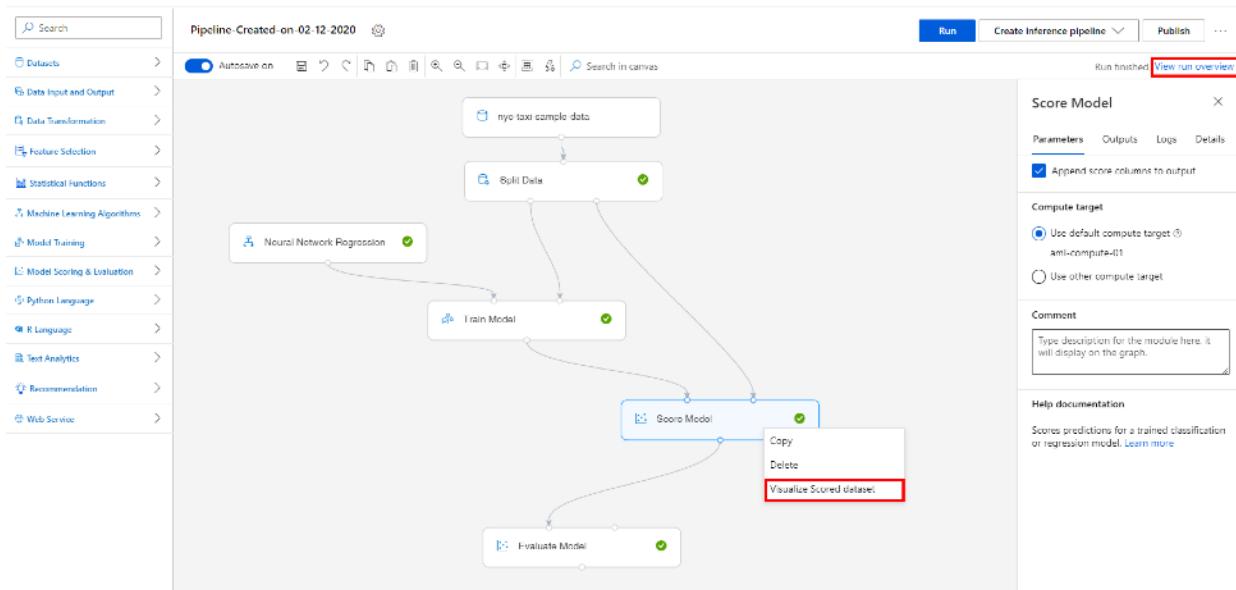


- Wait for pipeline run to complete. It will take around **8 minutes** to complete the run.
- While you wait for the model training to complete, you can learn more about the training algorithm used in this lab by selecting [Neural Network Regression module](#).

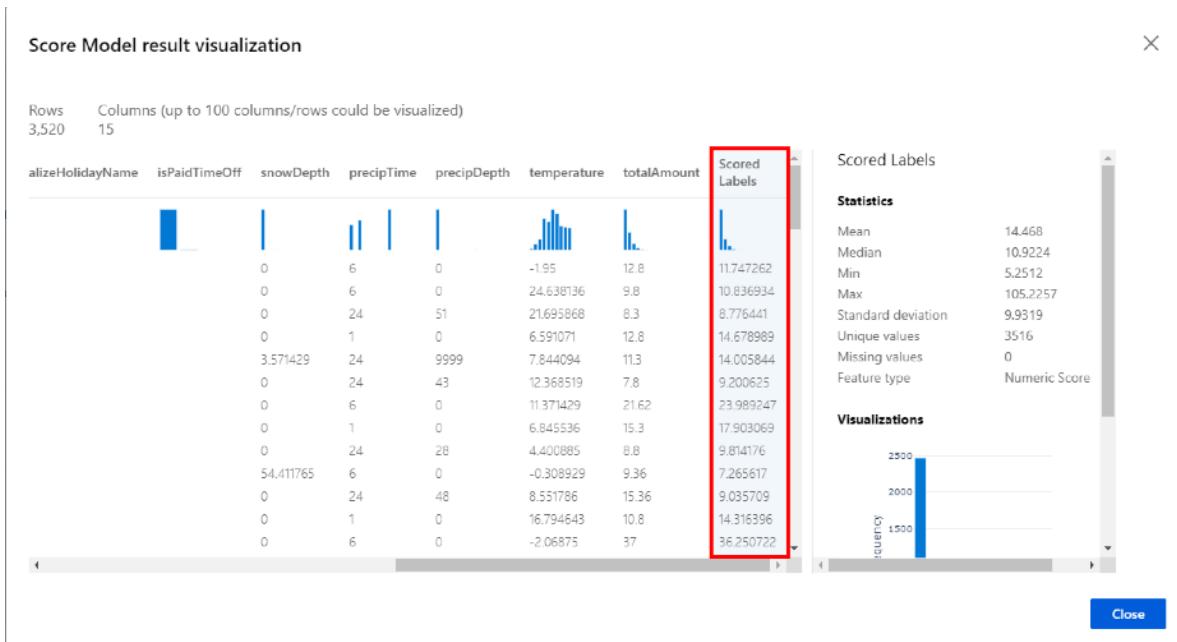
Exercise 4: Visualize Training Results

Task 1: Visualize the Model Predictions

- Select **Score Model, Outputs, Visualize** to open the [Score Model result visualization](#) dialog or just simply right-click the **Score Model** module and select **Visualize Scored Dataset**.

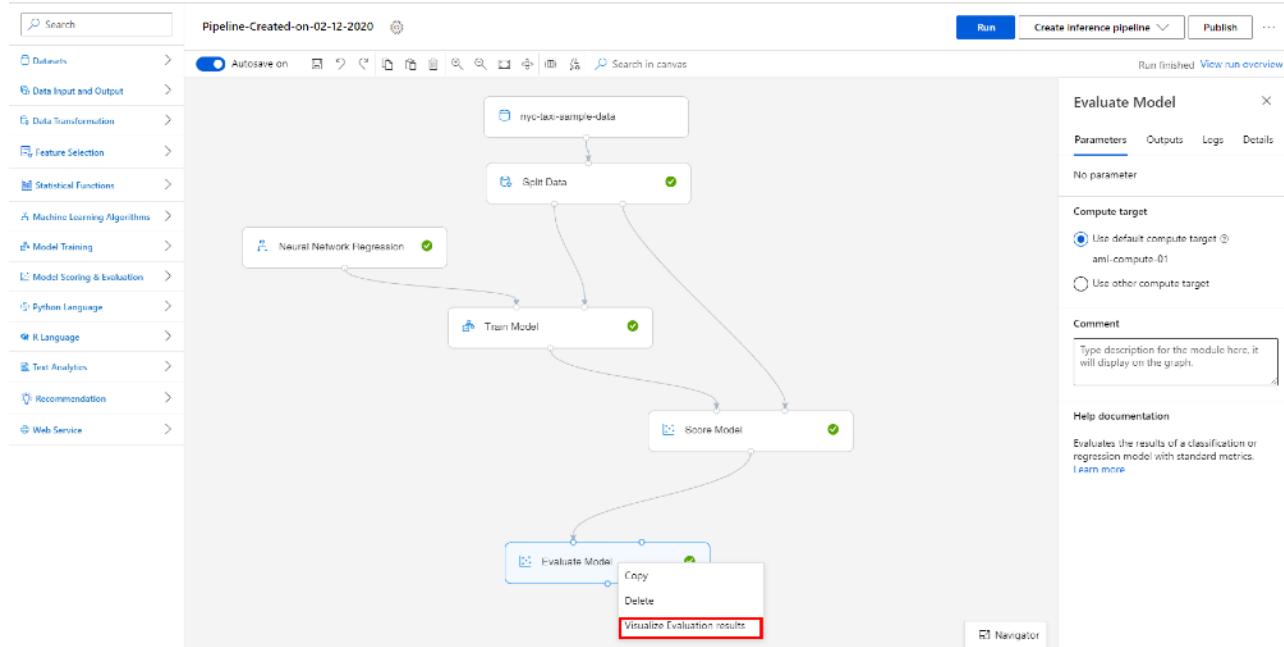


- Observe the predicted values under the column **Scored Labels**. You can compare the predicted values (**Scored Labels**) with actual values (**totalAmount**).

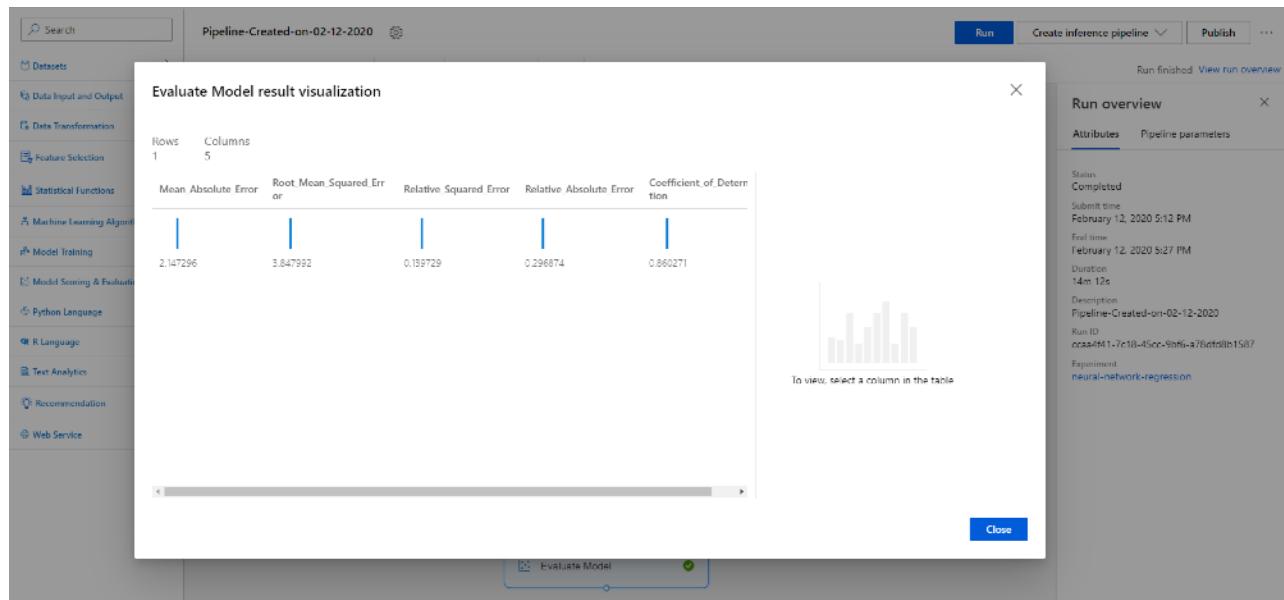


Task 2: Visualize the Evaluation Results

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



2. 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 neural net model using the prebuilt Neural Network Regression module 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.

Lab 15 - 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.

Overview

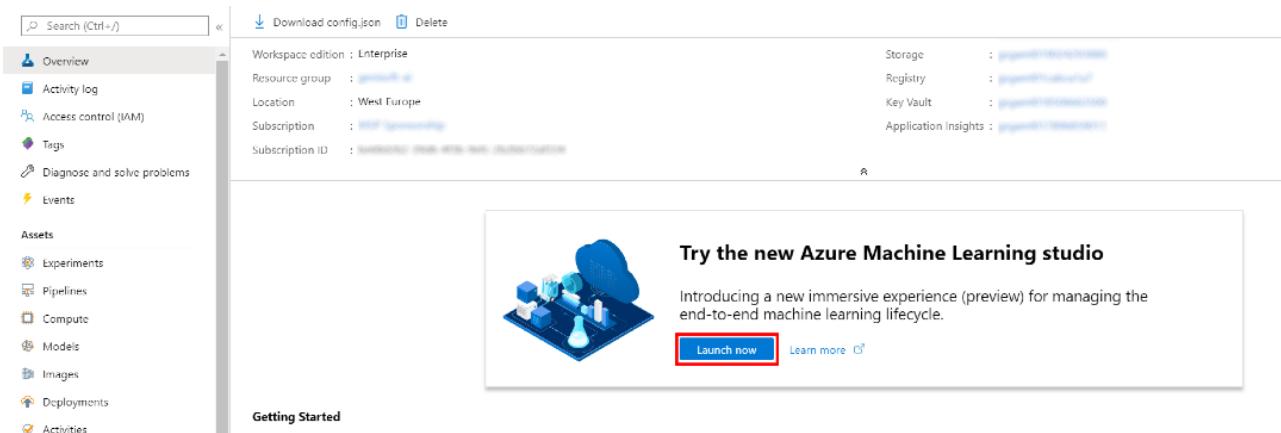
In this lab, we make use of the Train SVD Recommender module available in Azure Machine Learning designer (preview), to train a movie recommender engine. We use the collaborative filtering approach: the model learns from a collection of ratings made by users on a subset of a catalog of movies. Two open 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](#).

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](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

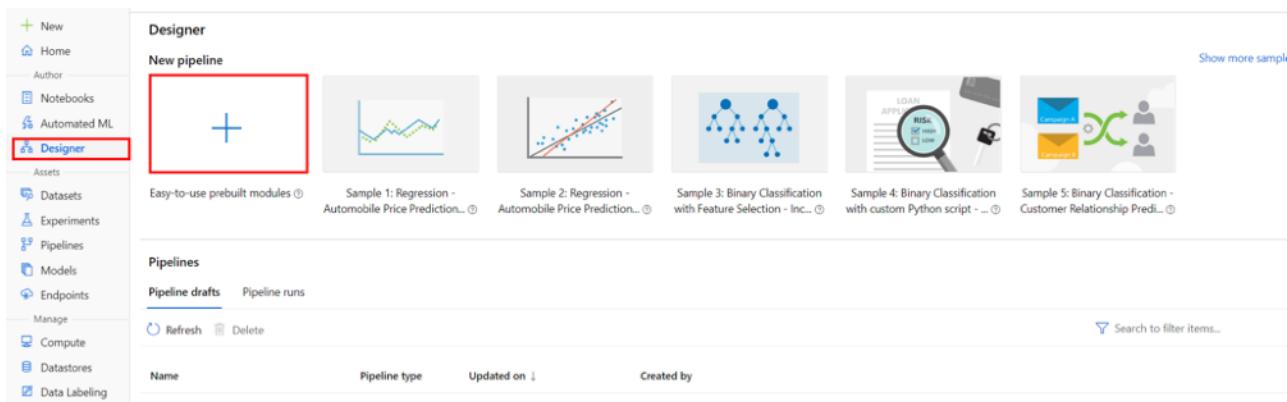
The screenshot shows the "Welcome to the studio!" screen. It has three dropdown menus:

- Switch directory:** The dropdown is set to "Udacity".
- Subscription:** The dropdown is set to "Azure Sponsorship - Udacity -04".
- Machine learning workspace:** The dropdown is set to "quick-starts-ws-190124". Below it, two other options are listed: "quick-starts-ws-190124" and "aml-quickstarts-190124", both associated with "southcentralus".

A blue "Get started" button is located at the bottom of the workspace dropdown.

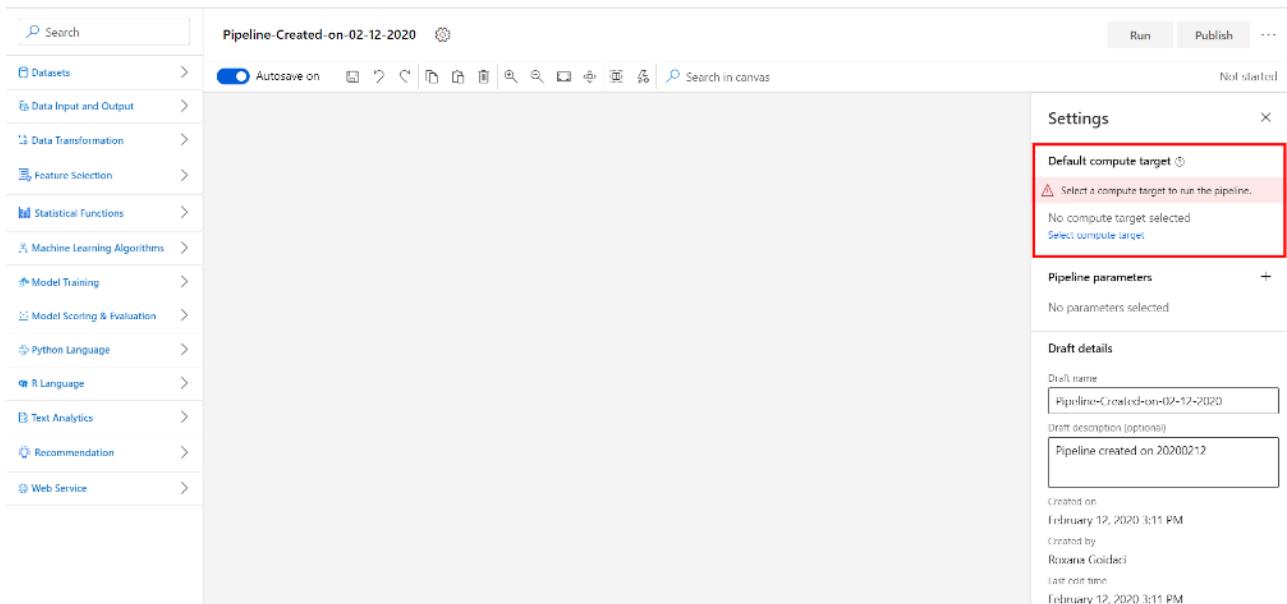
For the directory, select **Udacity** and for the subscription, select **Azure Sponsorship**. For the machine learning workspace, you may see multiple options listed. **Select any of these** (it doesn't matter which) and then click **Get started**.

4. 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**.



2. 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.

Search Pipeline-Created-on-02-12-2020 ⚙️

Run Publish ⋮

Datasets > Autosave on Not started

Data Input and Output >

Data Transformation >

Feature Selection >

Statistical Functions >

Machine Learning Algorithms >

Model Training >

Model Scoring & Evaluation >

Python Language >

R Language >

Text Analytics >

Recommendation >

Web Service >

Set up compute target

Select existing Create new

Existing compute target(s)

Refresh

Compute target name	Number of avail...	Region	Status
aml-compute-01	0 / 1	westeurope	succeeded

< Prev Next >

Save Cancel

Settings

Default compute target ⚙️

Select a compute target to run the pipeline

No compute target selected Select compute target

Pipeline parameters +

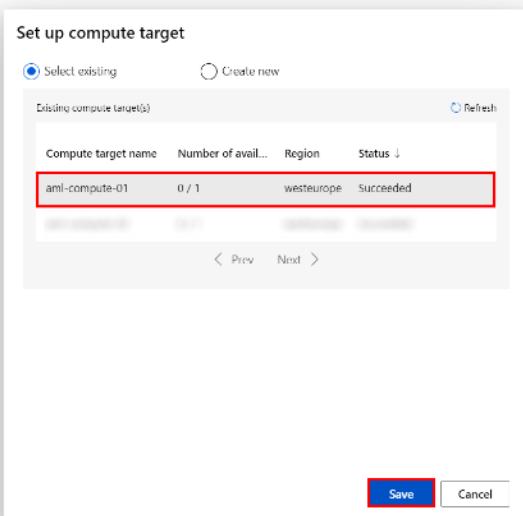
No parameters selected

Draft details

Draft name Pipeline-Created-on-02-12-2020

Draft description (optional) Pipeline created on 20200212

Created on February 12, 2020 3:11 PM



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.

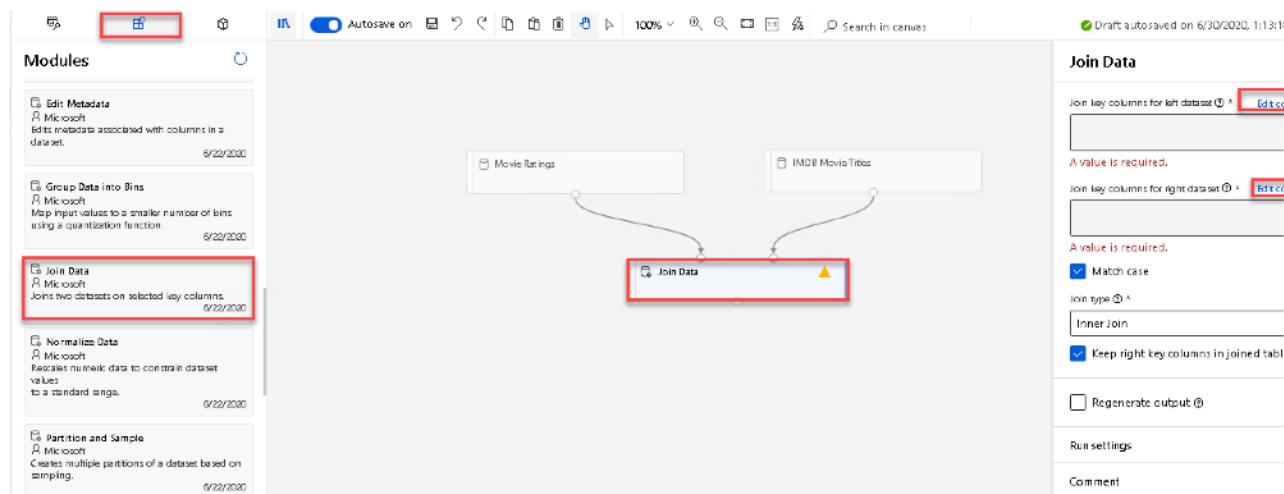
The screenshot shows the Azure Machine Learning Studio interface. On the left, there is a 'Datasets' pane containing a list of datasets. One dataset, 'Movie Ratings', is highlighted with a red box. On the right, there is a detailed view of the 'Movie Ratings' dataset, showing its parameters and outputs. The 'Parameters' tab is selected, displaying details like 'Dataset name: Movie Ratings', 'Datasource type: GlobalDataset', and 'Description: Movie Ratings'. The 'Outputs' tab is also visible. At the top right, there are 'Submit' and 'Publish' buttons, and a status message indicating the draft was autosaved on 6/30/2020, 1:11.

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

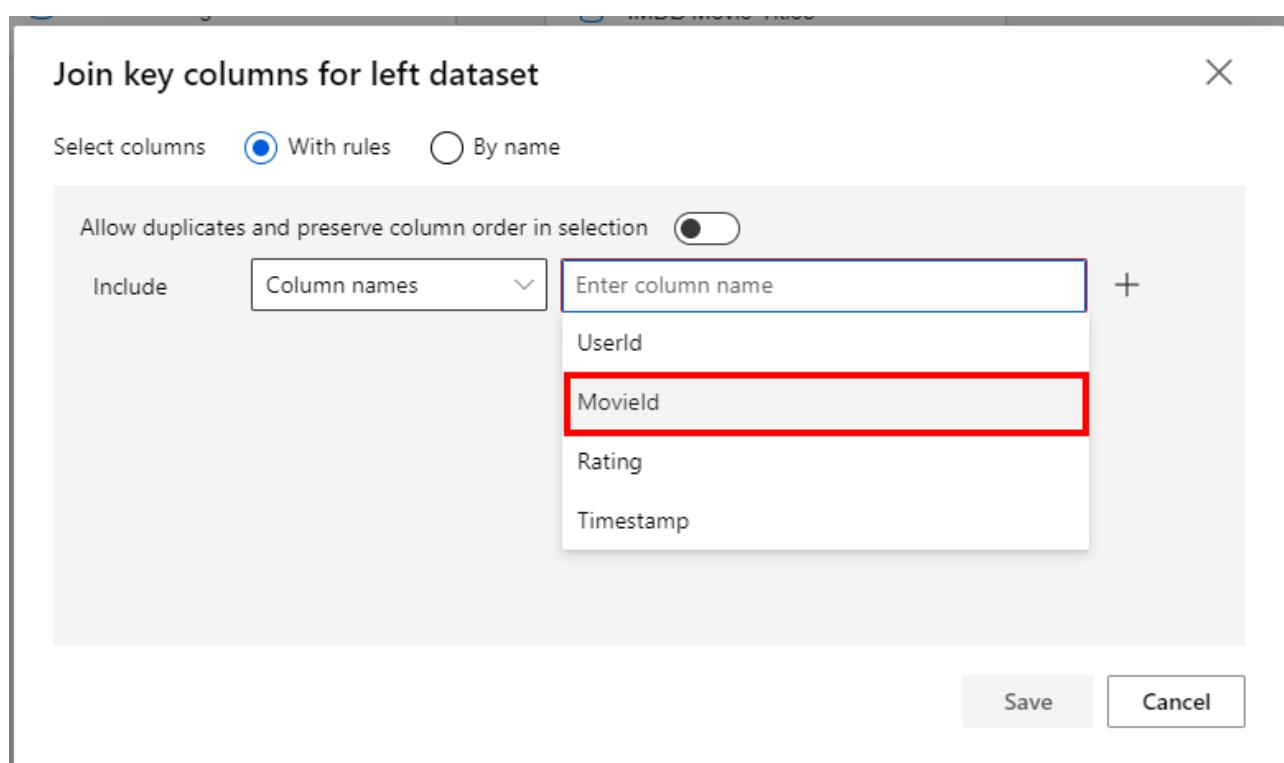
This screenshot shows the same interface as the previous one, but with a different dataset selected. The 'IMDB Movie Titles' dataset is now highlighted with a red box in the 'Datasets' pane. The detailed view on the right is still showing the 'Movie Ratings' dataset, but the status bar at the bottom right indicates the draft was autosaved on 6/30/2020, 1:35 C.

Task 4: Join the two datasets on Movie ID

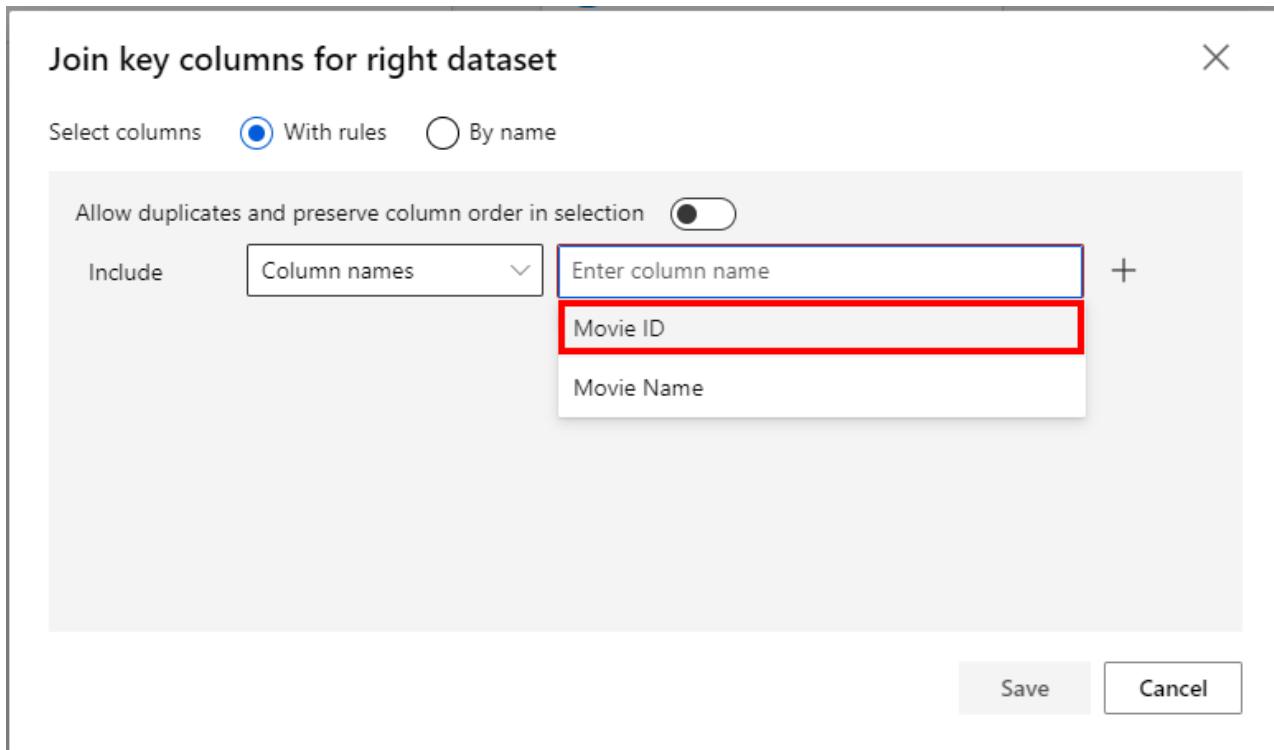
1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Join Data** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the output of the **Movie Ratings** module to the first input of the **Join Data** module.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



2. Select the **Join Data** module.
3. 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.



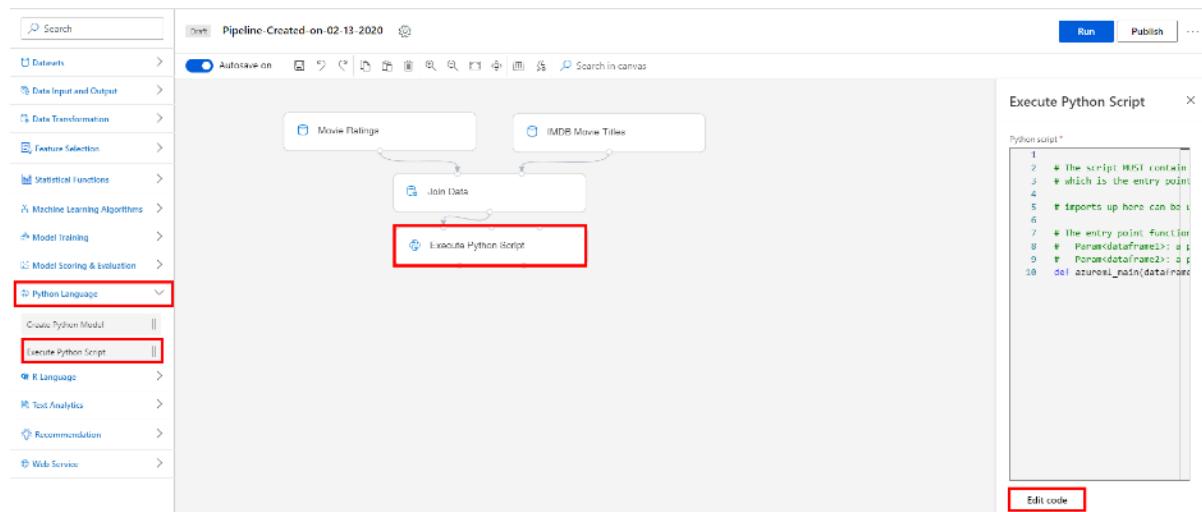
4. Select the **Edit column** link to open the **Join key columns for right dataset** editor. Select the **Movie ID** column in the **Enter column name** field.



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

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



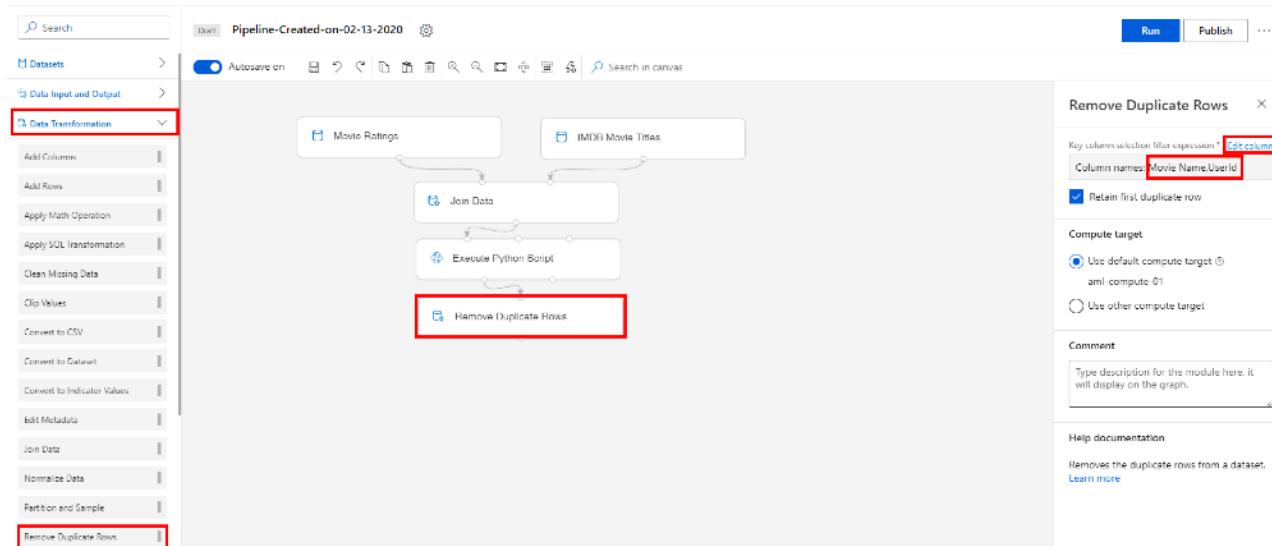
2. Select **Edit code** to open the **Python script** editor, clear the existing code and then enter the following lines of code to select the UserId, Movie Name, Rating columns from the joined dataset. Please ensure that there is no indentation for the first line and the second and third lines are indented.

```
def azureml_main(dataframe1 = None, dataframe2 = None):
    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.

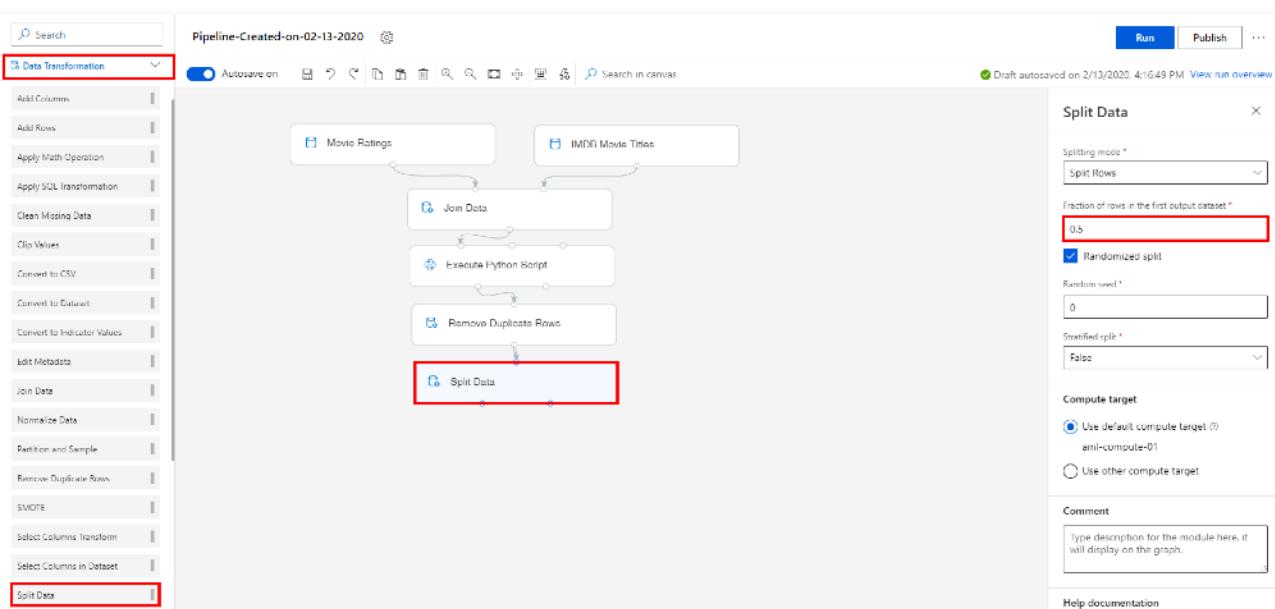
Task 6: Remove duplicate rows with same Movie Name and UserId

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**.



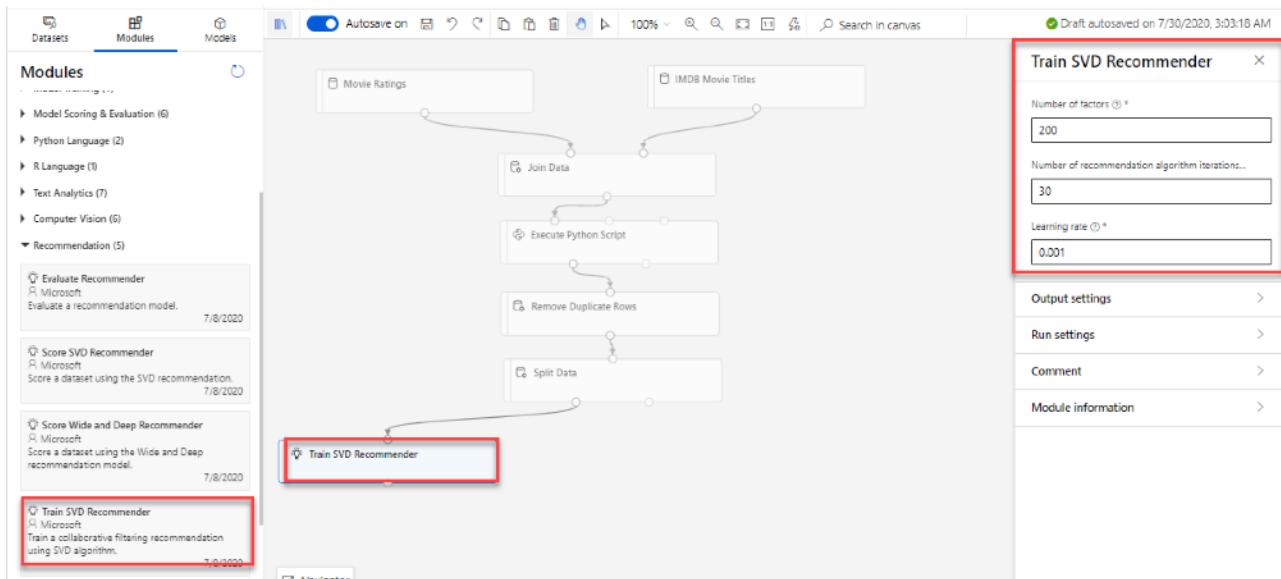
Task 7: Split the dataset into training set (0.5) and test set (0.5)

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.

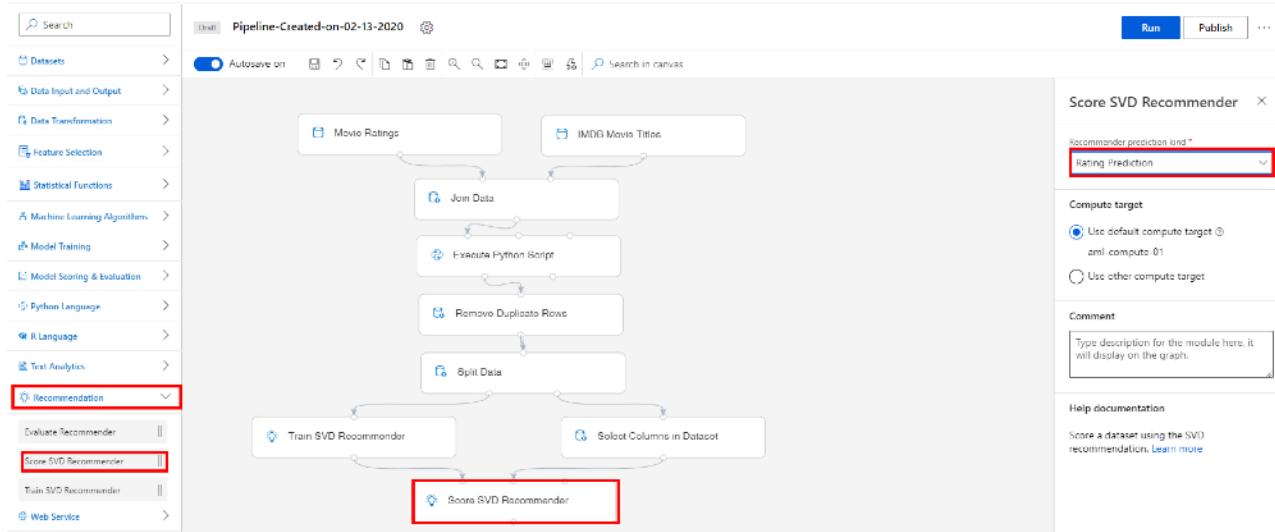


Task 9: Select Columns UserId, Movie Name from the test set

- Select **Data Transformation** section in the left navigation. Follow the steps outlined below:
 - Select the **Select Columns in Dataset** prebuilt module.
 - Drag and drop the selected module on to the canvas.
 - Connect the **Split Data** second output to the input of the **Select columns in Dataset** module.
 - 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**.

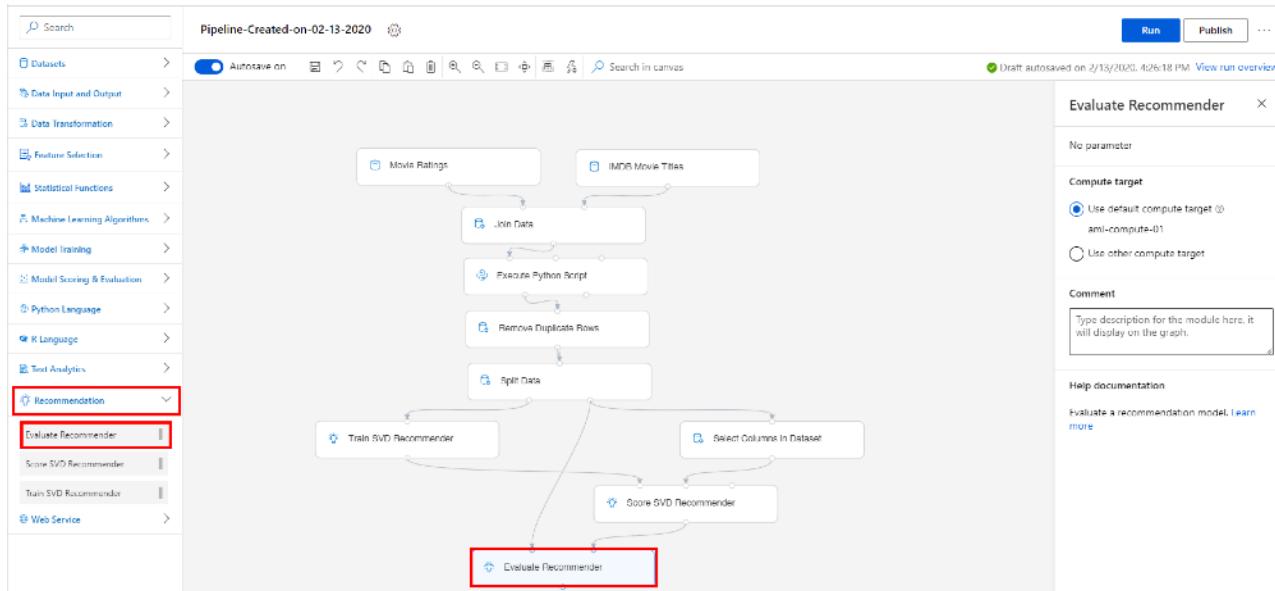
Task 10: Configure the Score SVD Recommender

- Select **Recommendation** section in the left navigation. Follow the steps outlined below:
 - Select the **Score SVD Recommender** prebuilt module.
 - Drag and drop the selected module on to the canvas
 - 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.
 - 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.
 - Select the **Score SVD Recommender** module on the canvas.
 - 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

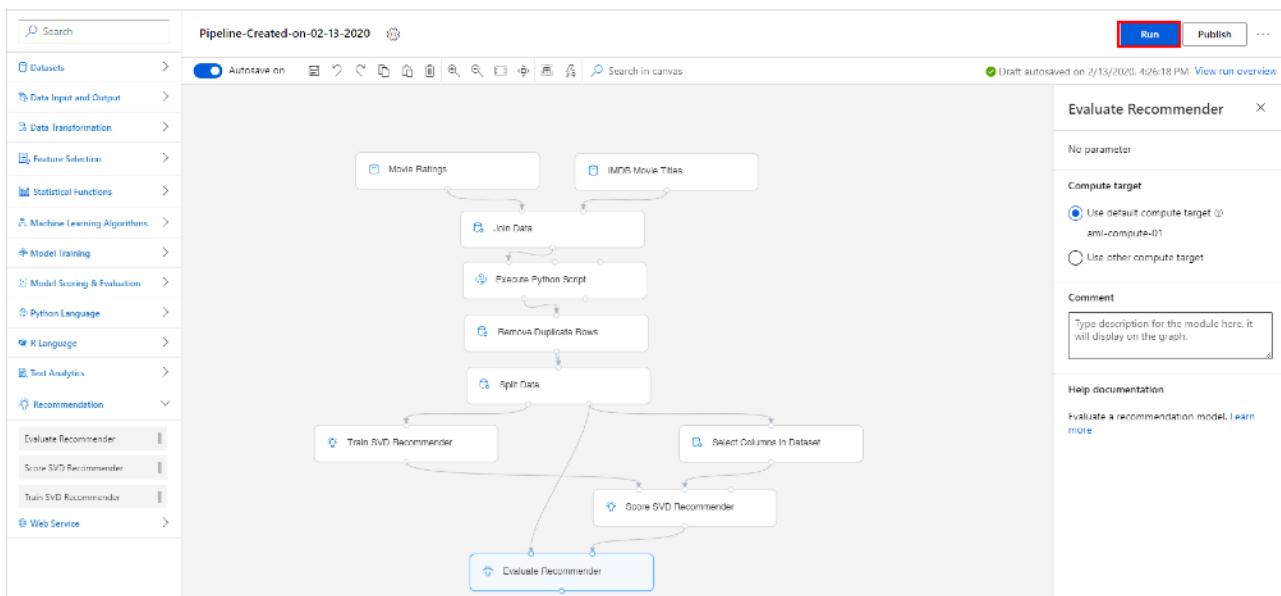
1. Select **Recommendation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Recommender** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the **Score SVD Recommender** module to the second input of the **Evaluate Recommender** 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

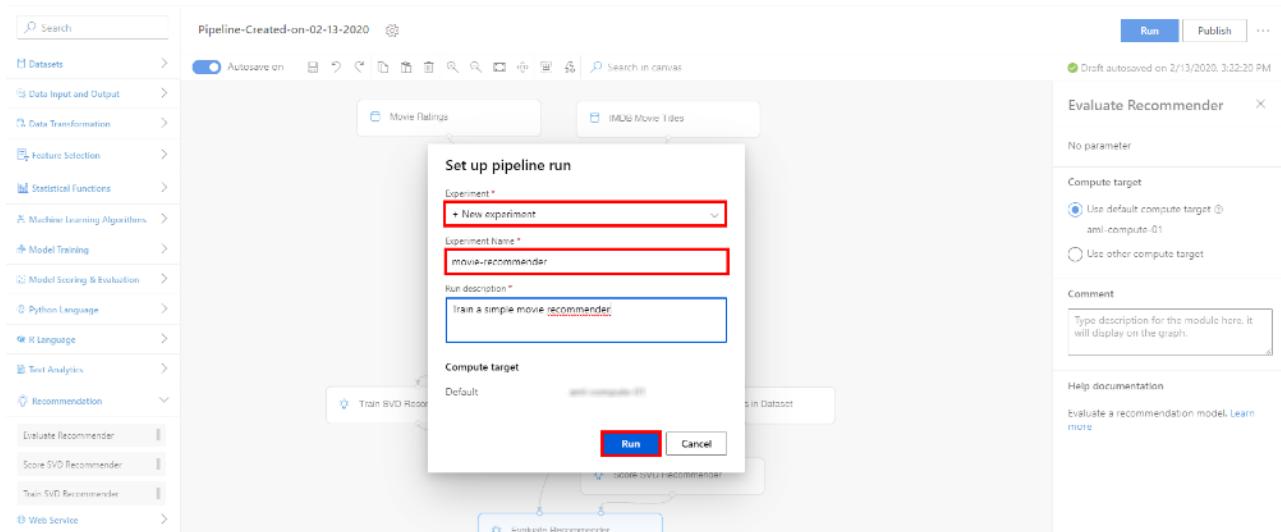
Task 1: Create Experiment and Submit Pipeline

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**.

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

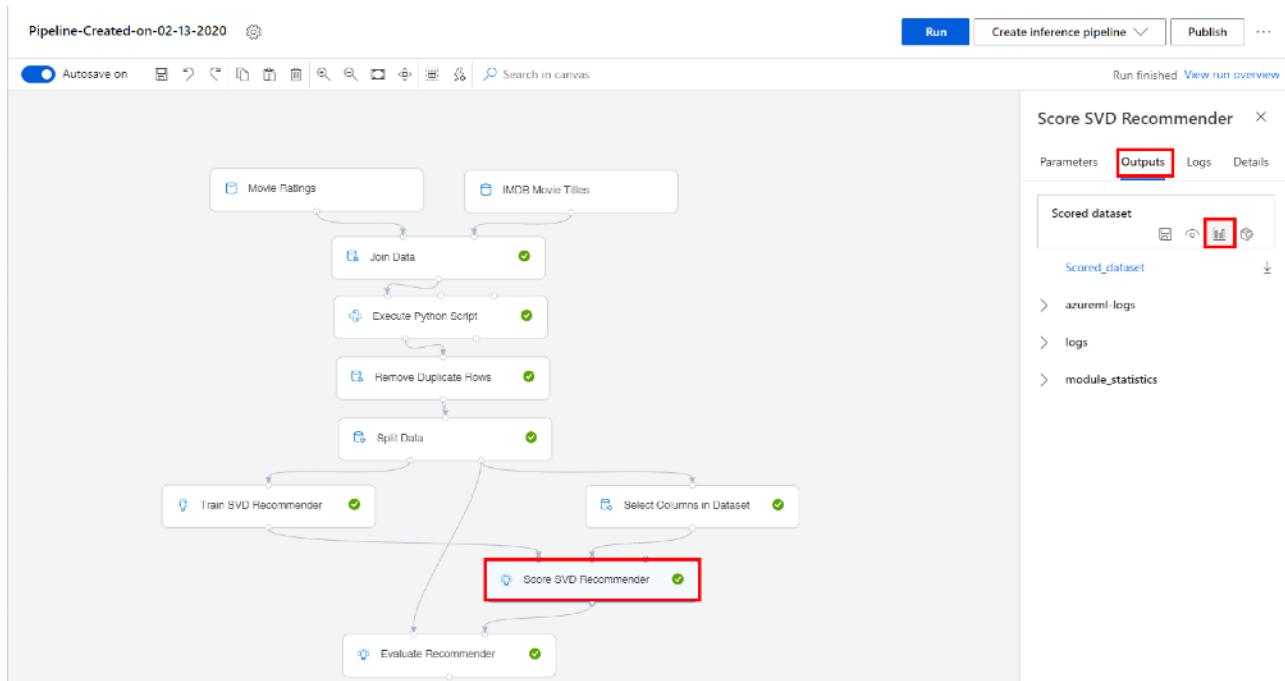


3. Wait for pipeline run to complete. It will take around **20 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the SVD algorithm used in this lab by selecting **Train SVD Recommender**.

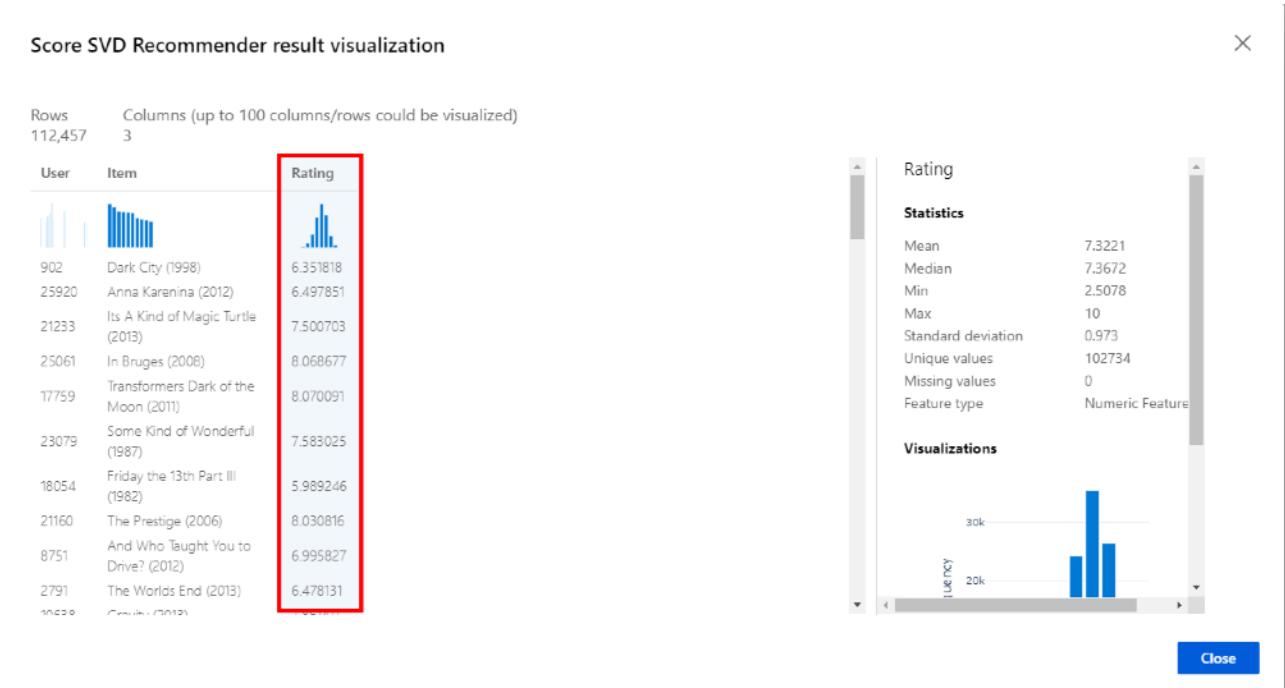
Exercise 3: Visualize Scoring Results

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**.

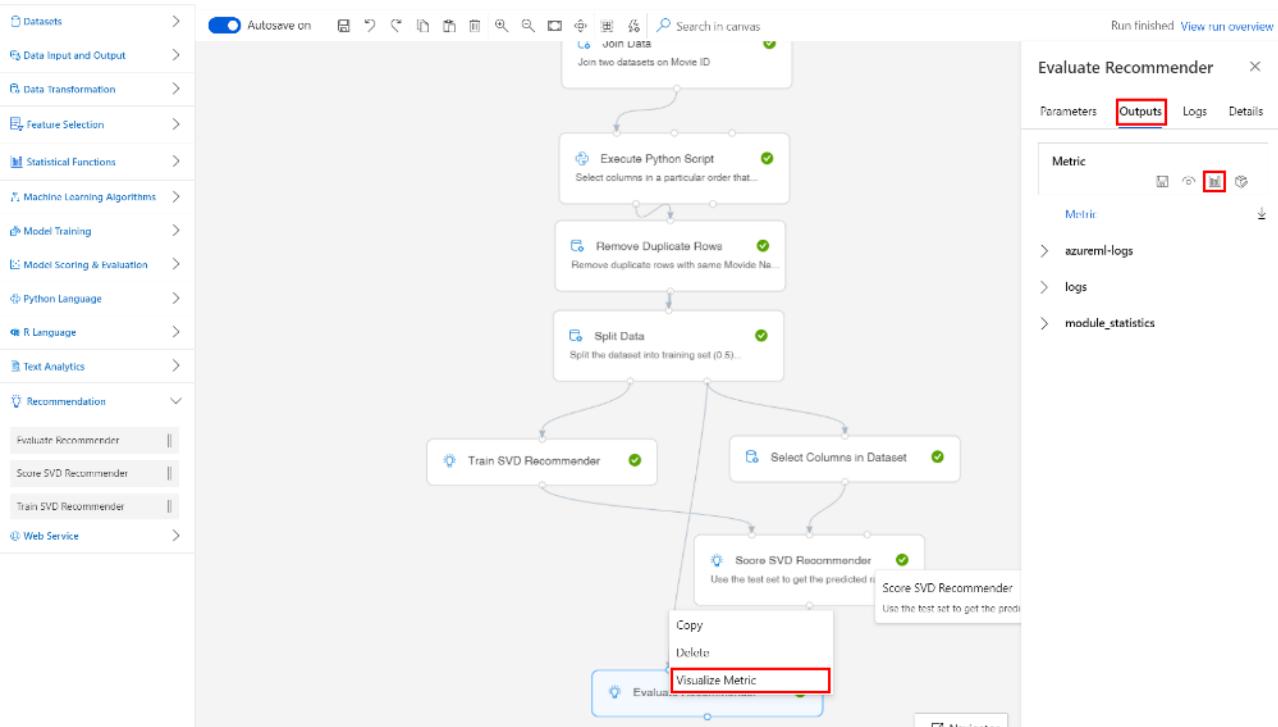


2. Observe the predicted values under the column **Rating**.

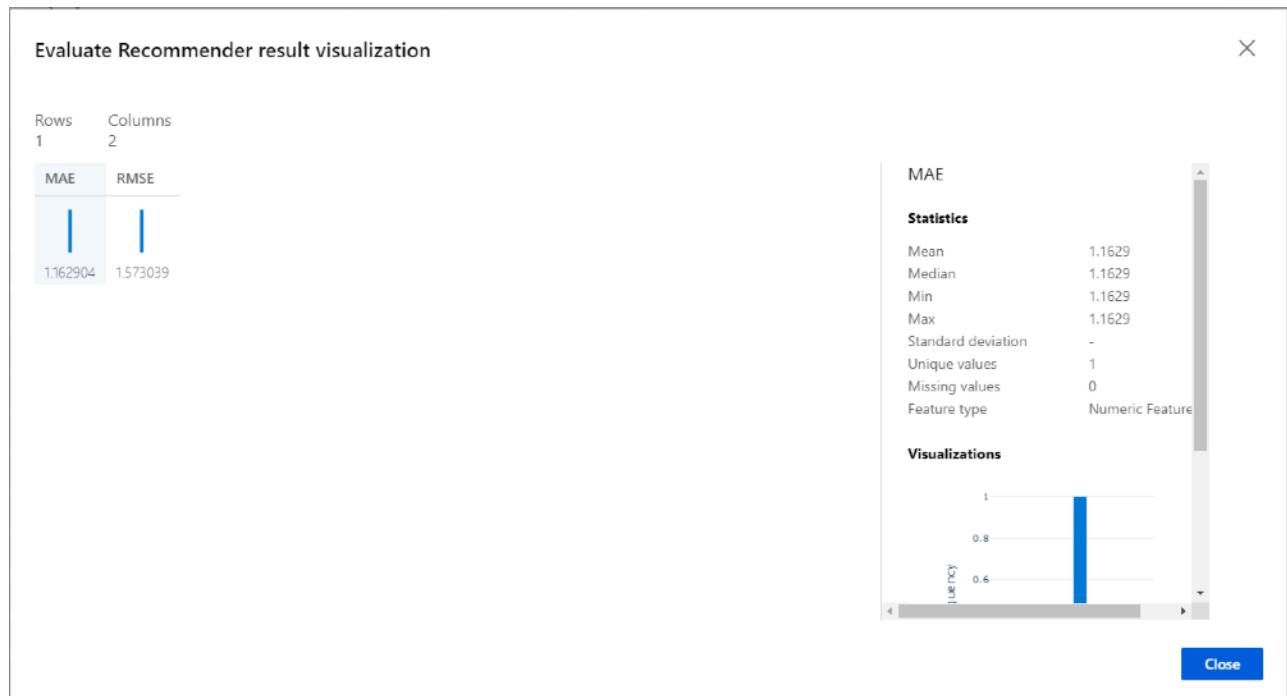


Task 2: Visualize the Evaluation Results

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**.



2. 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.

Lab 16 - Train a simple text classifier

In text classification scenarios, the goal is to assign a piece of text, such as a document, a news article, a search query, an email, a tweet, support tickets, customer feedback, user product review, to predefined classes or categories. Some examples of text classification applications are: categorizing newspaper articles into topics, organizing web pages into hierarchical categories, spam email filtering, sentiment analysis, predicting user intent from search queries, support tickets routing, and customer feedback analysis.

Overview

In this lab we demonstrate how to use text analytics modules available in Azure Machine Learning designer (preview) to build a simple text classification pipeline. We will create a training pipeline and initialize a **multiclass logistic regression classifier** to predict the company category with Wikipedia SP 500 dataset derived from Wikipedia. The dataset manages articles of each S&P 500 company. Before uploading to Azure Machine Learning designer, the dataset was processed as follows: extracted text content for each specific company, removed wiki formatting, removed non-alphanumeric characters, converted all text to lowercase, known company categories added. Articles could not be found for some companies, so that's why the number of records is less than 500.

Exercise 1: Create New Training Pipeline

Task 1: Open Pipeline Authoring Editor

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

The screenshot shows the Azure Machine Learning workspace Overview page. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets (Experiments, Pipelines, Compute, Models, Images, Deployments, Activities), and Getting Started. The main area displays workspace details: Edition (Enterprise), Resource group (generated-ml), Location (West Europe), Subscription (mlt-quickstarts), and Subscription ID (generated-mlt-quickstarts-190124). Below this is a 'Try the new Azure Machine Learning studio' card with a 'Launch now' button, which is also highlighted with a red box.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

The screenshot shows the 'Welcome to the studio!' configuration screen. It has three dropdown menus: 'Switch directory' (set to 'Udacity'), 'Subscription' (set to 'Azure Sponsorship - Udacity -04'), and 'Machine learning workspace' (set to 'quick-starts-ws-190124'). The 'Machine learning workspace' dropdown shows other options: 'quick-starts-ws-190124' (selected), 'quick-starts-ws-190124' (disabled), and 'aml-quickstarts-190124'. A 'Get started' button is at the bottom.

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**.

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

Task 2: Setup Compute Target

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

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

The screenshot shows the 'Set up compute target' dialog box. It contains a table with one row: 'aml-compute-01' (Compute target name), '0 / 1' (Number of avail...), 'westeurope' (Region), and 'Succeeded' (Status). The entire row is highlighted with a red box. At the bottom of the dialog are two buttons: 'Save' (highlighted with a red box) and 'Cancel'.

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 Wikipedia SP 500 Sample Datasets

1. Select **Datasets** section in the left navigation. Next, select **Samples, Wikipedia SP 500 Dataset** and drag and drop the selected dataset on to the canvas.

The screenshot shows the Azure ML Studio interface. On the left, there is a 'Datasets' list with several datasets listed: CRM Dataset Shared, CRM Upselling Labels Shared, Flight Delays Data, German Credit Card UCI dataset, IMDB Movie Titles, Movie Ratings, Weather Dataset, and Wikipedia SP 500 Dataset. The 'Wikipedia SP 500 Dataset' item is highlighted with a red box. On the right, there is a detailed view of the 'Wikipedia SP 500 Dataset' with tabs for 'Parameters' and 'Outputs'. The 'Parameters' tab shows the dataset name as 'Wikipedia SP 500 Dataset', datasource type as 'GlobalDataset', and data type as 'DataFrameDirectory'. The 'Outputs' tab shows the datastore name as 'azureml_globaldatasets' and relative path as 'GenericCSV/Wikipedia_SP_500_Dataset'. Both the 'Datasets' list and the 'Wikipedia SP 500 Dataset' details pane are highlighted with red boxes.

Task 4: Preprocess text for following steps

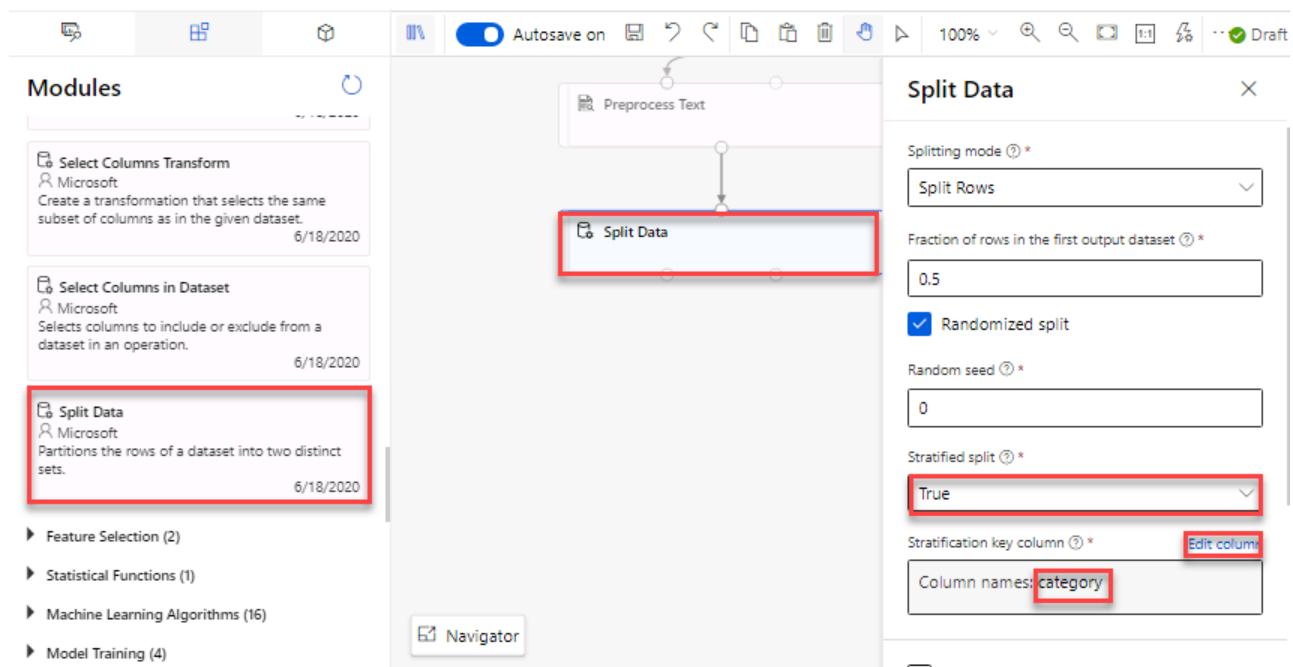
1. Select **Text Analytics** section in the Modules . Follow the steps outlined below:
 1. Select the **Preprocess Text** prebuilt module.
 2. Drag and drop the selected module on to the canvas.
 3. Connect the output of the **Wikipedia SP 500** module to the input of the **Preprocess Text** module. The dataset input of this prebuilt module needs to be connected to a dataset that has at least one column containing text.
 4. Select the language from the **Language** dropdown list: **English**.
 5. Select the **Edit column** link to open the **Text column to clean** editor. Select the **Text** column in the **Enter column name** field.
 6. Leave all the other options checked, as in the default configuration of the **Preprocess Text** module.

The screenshot shows the Azure Machine Learning Studio interface. On the left, there is a list of modules under the 'Modules' section. One module, 'Preprocess Text', is highlighted with a red box. In the center, there is a pipeline canvas with a 'Wikipedia SP 500 Dataset' module connected to a 'Preprocess Text' module. A connection line is visible between them. On the right, a 'Preprocess Text' configuration dialog is open. It shows the following settings:

- Language:** English
- Text column to clean:** Text (with an 'Edit column' link)
- Other options checked:** Expand verb contractions, Remove stop words, Use lemmatization, Detect sentences, Normalize case to lowercase, Remove numbers, Remove special characters, Remove duplicate characters, Remove email addresses

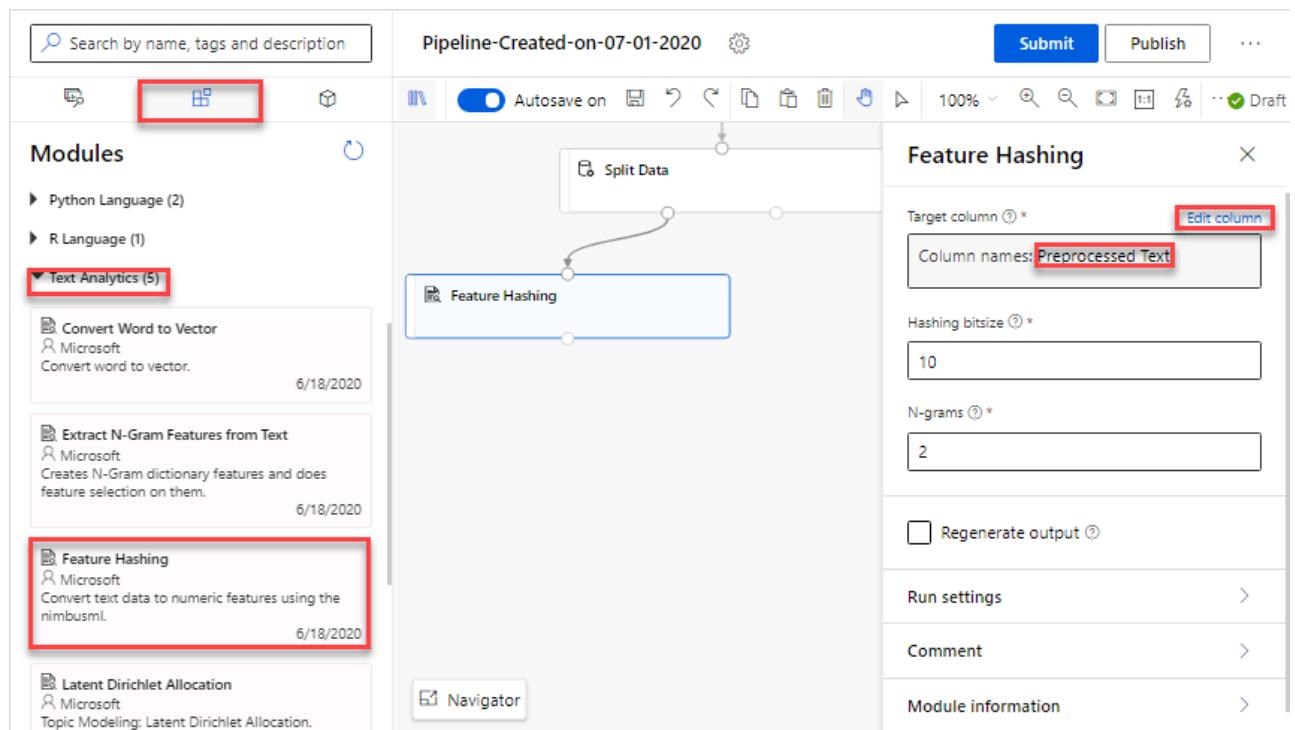
Task 5: Split the dataset into training set (0.5) and test set (0.5)

1. Select **Data Transformation** section in the Modules. 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. Stratified seed: **True**.
 5. Select the **Edit column** link to open the **Stratification key column** editor. Select the **Category** column in the **Enter column name** field.
 6. Connect the output of the **Preprocess Text** module to the input of the **Split Data** module.



Task 6: Convert the plain text of the articles to integers with Feature Hashing module, on the training set

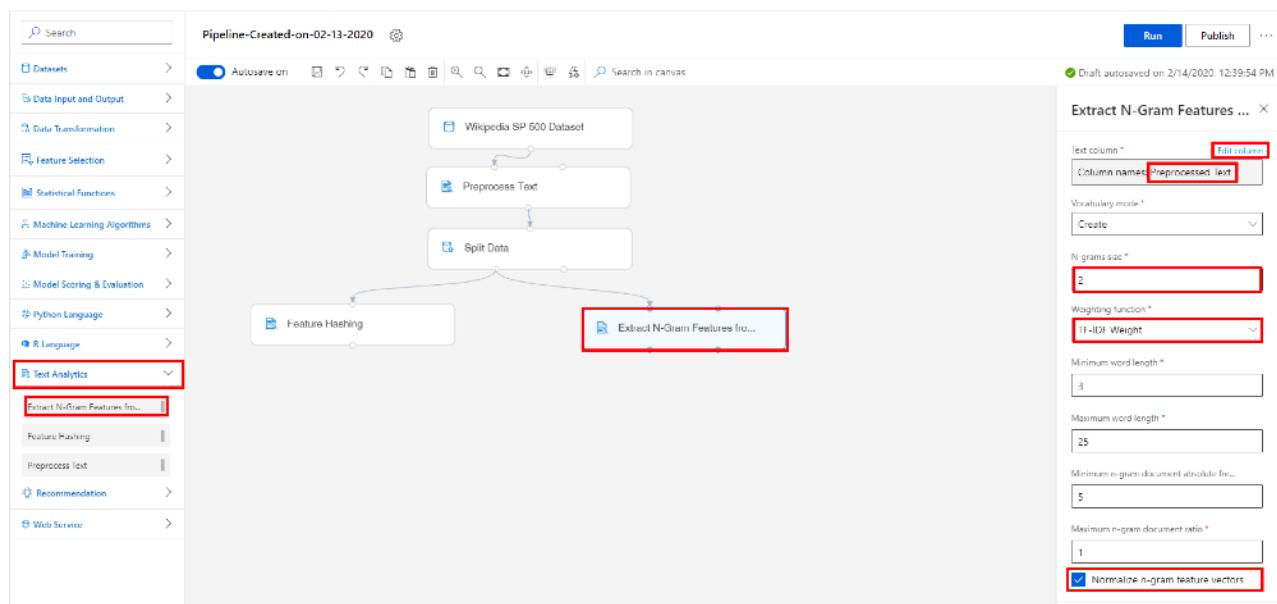
1. Select **Text Analytics** section in the left navigation. Follow the steps outlined below:
 1. Select the **Feature Hashing** prebuilt module.
 2. Drag and drop the selected module on to the canvas.
 3. Connect the first output of the **Split Data** to the input of the **Feature Hashing** module.
 4. Select the **Edit columns** link to open the **Target column** editor and then enter the **Preprocessed Text** column for the Target column field.
 5. Set **Hashing bitsize**: **10** and set the number of **N-grams**: **2**.



The goal of using feature hashing is to reduce dimensionality; also it makes the lookup of feature weights faster at classification time because it uses hash value comparison instead of string comparison.

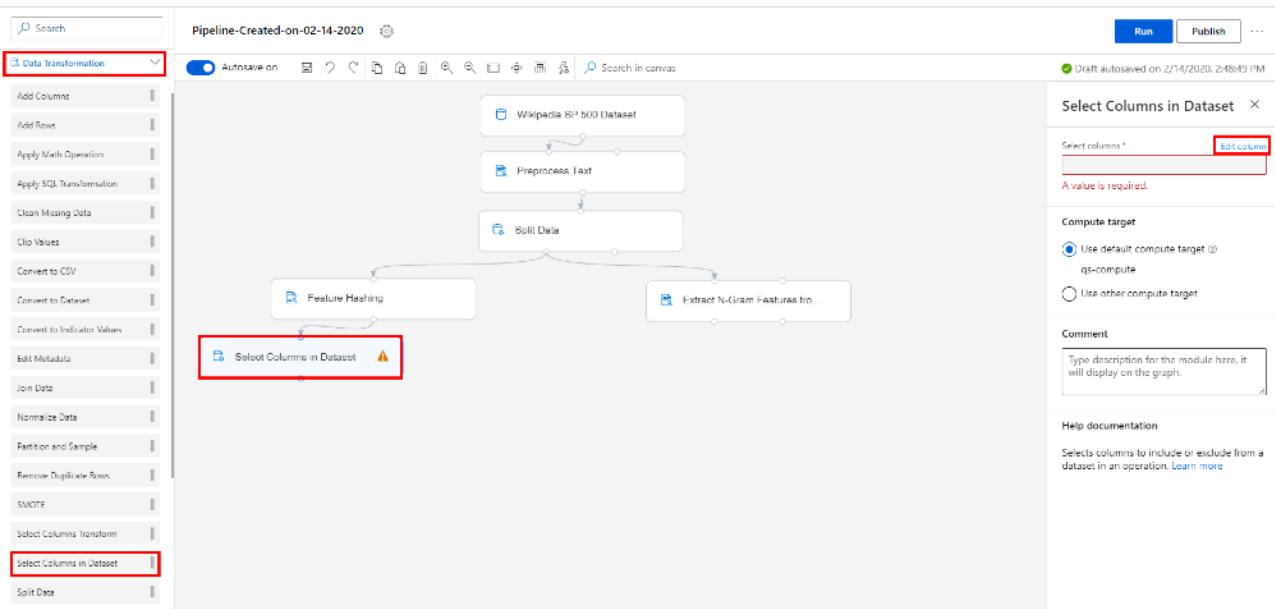
Task 7: Featurize unstructured text data with Extract N-Gram Feature from Text module, on the training set

1. Select **Text Analytics** section in the left navigation. Follow the steps outlined below:
 1. Select the **Extract N-Gram Feature from Text** prebuilt module.
 2. Drag and drop the selected module on to the canvas.
 3. Connect the first output of the **Split Data** to the input of the **Extract N-Gram Feature from Text** module.
 4. Select the **Edit columns** link to open the **Text column** editor and then enter the **Preprocessed Text** column for the Target column field.
 5. Leave default selection of **Vocabulary mode** to **Create** to indicate that you're creating a new list of n-gram features.
 6. Set **N-grams size**: **2** (which is the maximum size of the n-grams to extract and store)
 7. Select **Weighting function**: **TF-IDF Weight**. (This function calculates a term frequency/inverse document frequency score and assigns it to each n-gram. The value for each n-gram is its TF score multiplied by its IDF score.)
 8. Check the option to **Normalize n-gram feature vectors**. If this option is enabled, each n-gram feature vector is divided by its L2 norm.

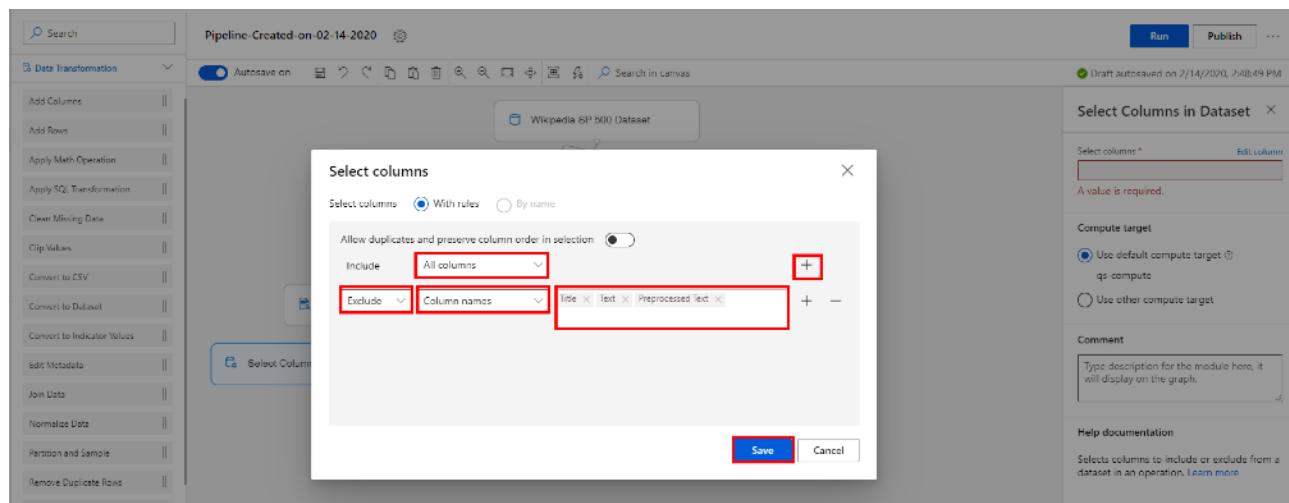


Task 8: Remove text columns from dataset

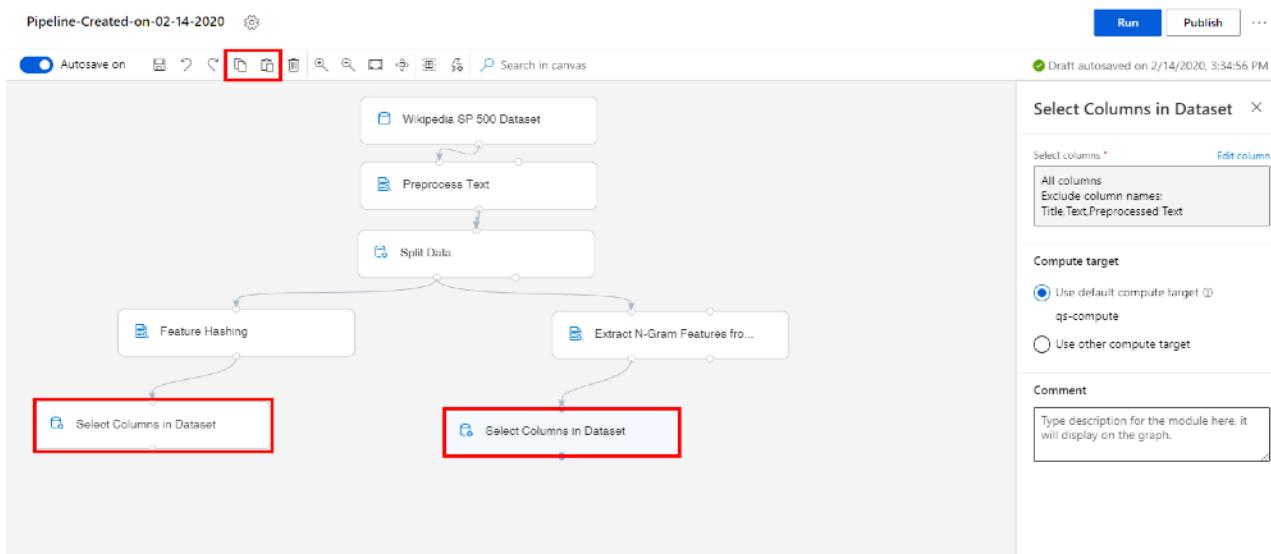
1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below to add two Select Columns in Dataset modules on both featurization branches:
 1. Select the **Select Columns in Dataset** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the input **Select Columns in Dataset** module to the **Feature Hashing** module output.



2. Select the **Edit columns** link to open the **Select columns** editor and then select **All columns**.
3. Click on the **+** icon to add another operation line and select the **Exclude, Column names** option. Enter to following list of columns to be excluded: **Title, Text, Preprocessed Text**. Select **Save** to close the editor.

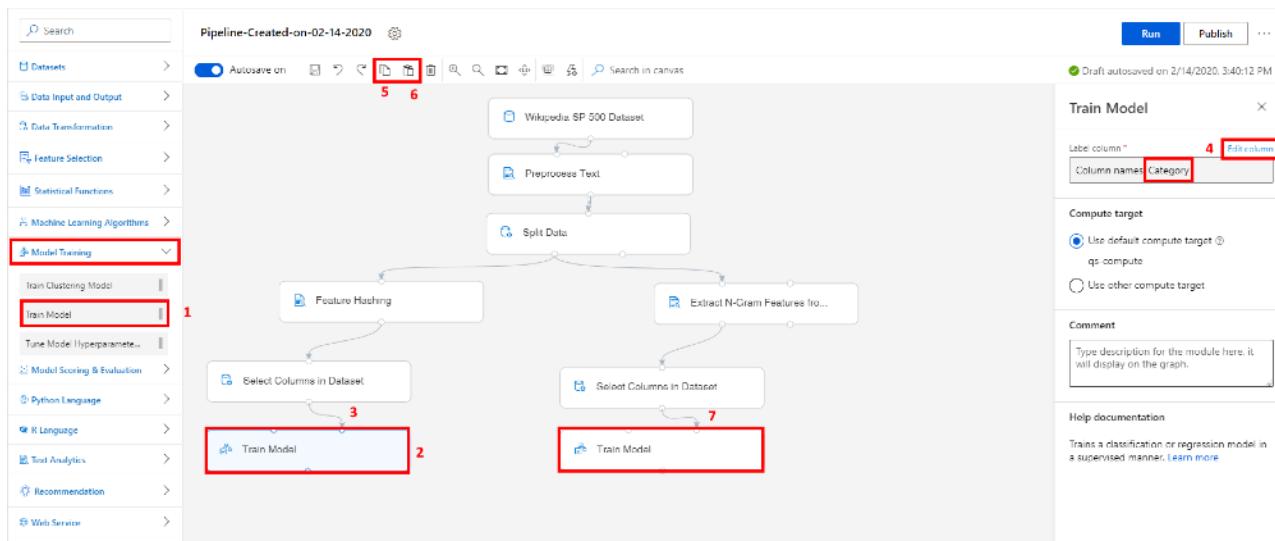


4. Select the **Select Columns in Dataset** module and click on the **Copy** icon and then **Paste** it on the canvas. Connect the copied module to the **Extract N-Gram Feature from Text** module.



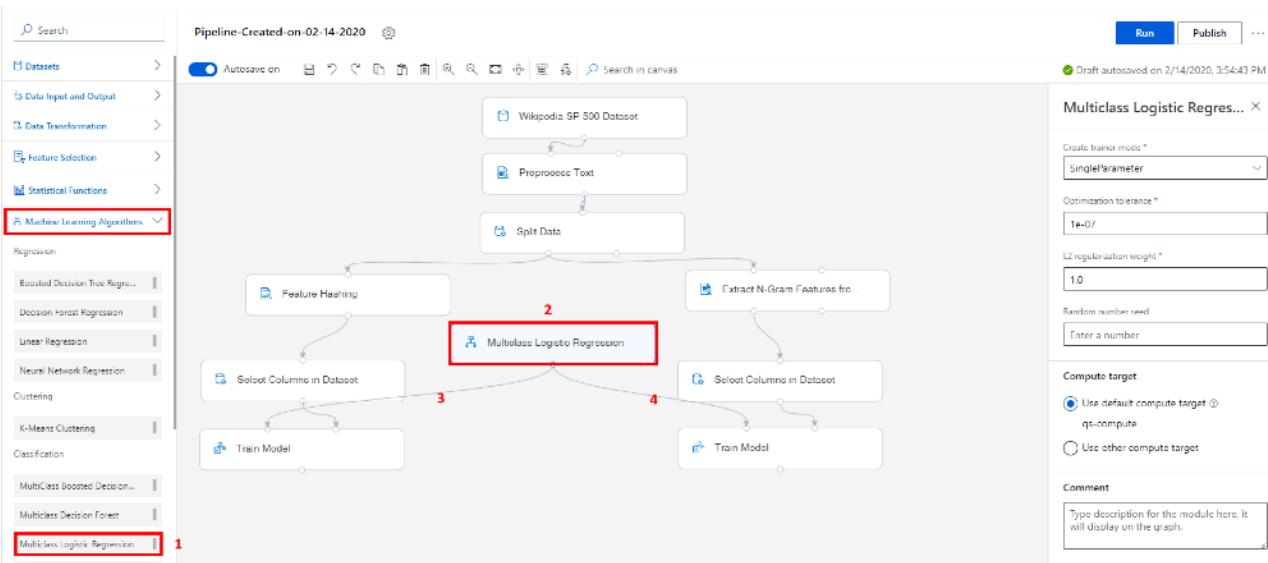
Task 9: Add the Train Model Modules

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
 1. Select the **Train Model** prebuilt module.
 2. Drag and drop the selected module on to the canvas
 3. Connect the first output of the **Select Columns in Dataset** module on the left of the canvas, to the second input (the Dataset input) of the newly added **Train Model** module.
 4. Select the **Edit columns** link to open the **Label column** editor and then enter the **Category** column. Select **Save** to close the editor.
 5. Select the **Train Model** module and click on the **Copy** icon.
 6. Click on the **Paste** icon to paste a second **Train Model** module on the canvas under the right branch of the pipeline tree.
 7. Connect this last **Train Model** module, by linking at the second input (the Dataset input), the output of the right most **Select Columns in Dataset** module on the canvas.



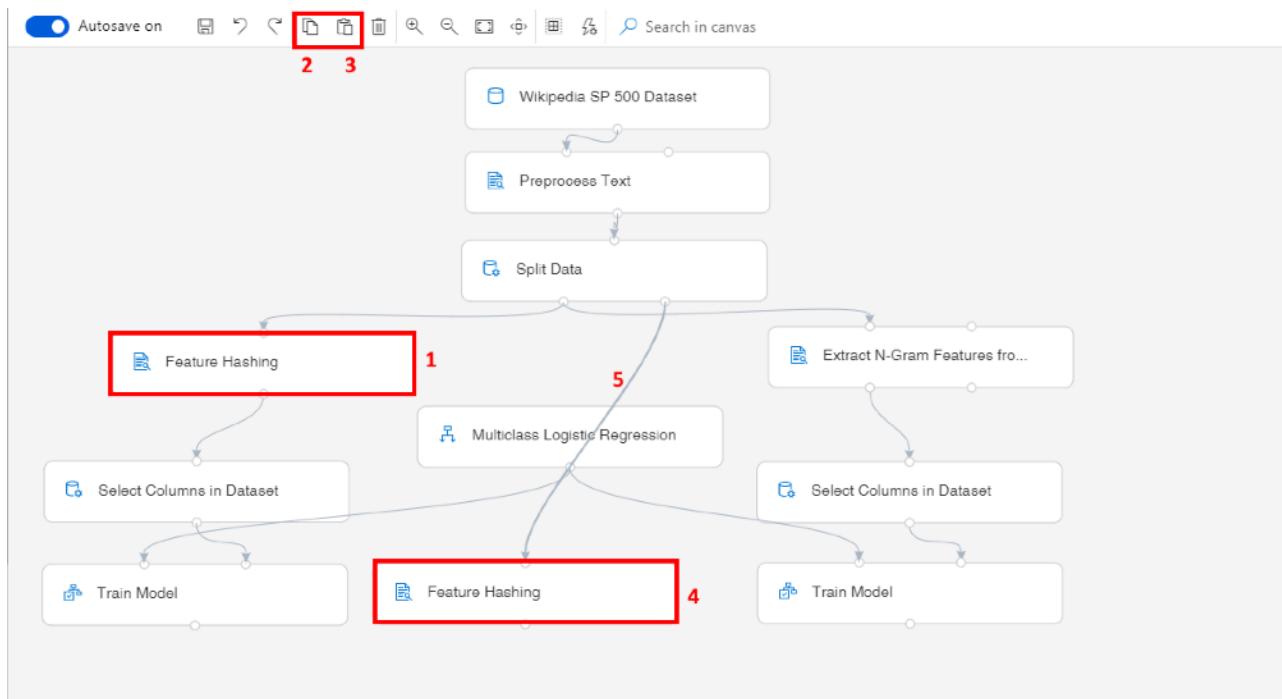
Task 10: Initialize Multiclass Logistic Regression Model

1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
 1. Select the **Multiclass Logistic Regression** prebuilt module, in the **Classification** category.
 2. Drag and drop the selected module on to the canvas
 3. Connect the output of the **Multiclass Logistic Regression** module to the first input of the left branch (the Feature Hashing approach) **Train Model** module.
 4. Connect the output of the **Multiclass Logistic Regression** module to the first input of the right branch (the N-Gram Features approach) **Train Model** module.



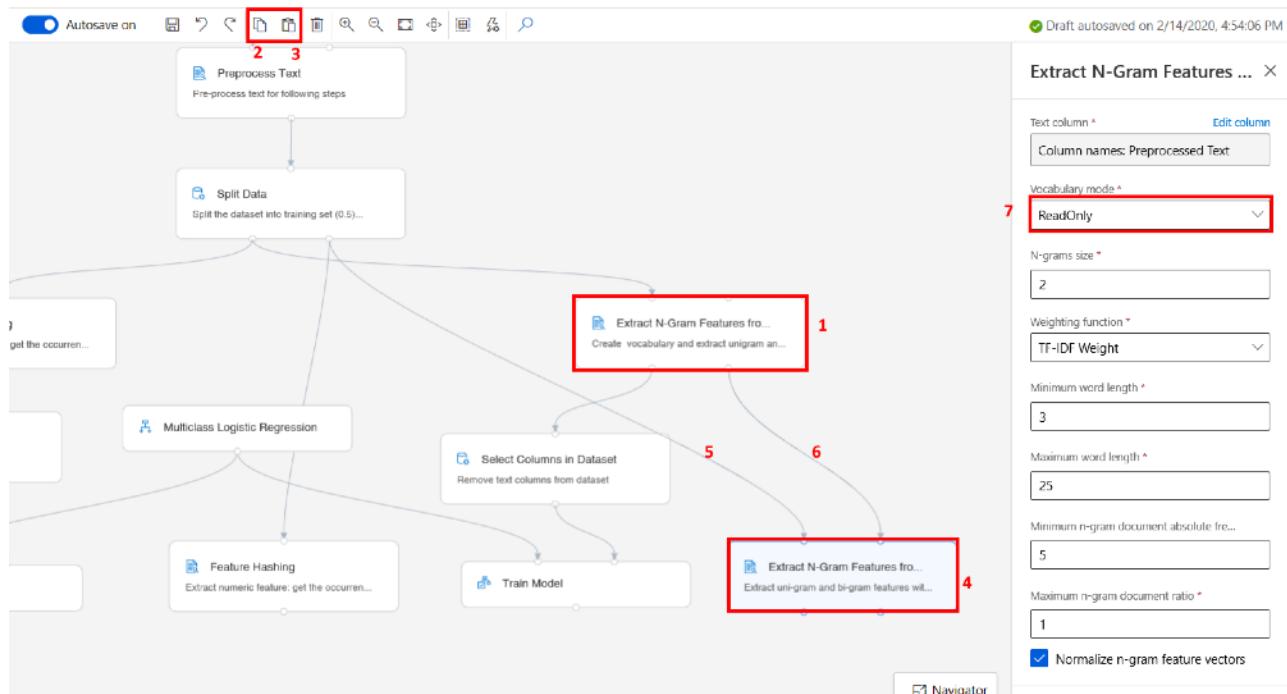
Task 11: Convert the plain text of the articles to integers with Feature Hashing module, on the test set

1. Select the existing **Feature Hashing** module.
2. Copy the **Feature Hashing** module using the **Copy** option from the top menu.
3. **Paste** the copied module on the canvas.
4. Position the new **Feature Hashing** module between the two **Train Model** modules on the canvas.
5. Connect the second output of the **Split Data**, the test set output, to the input of the newly added **Feature Hashing** module.



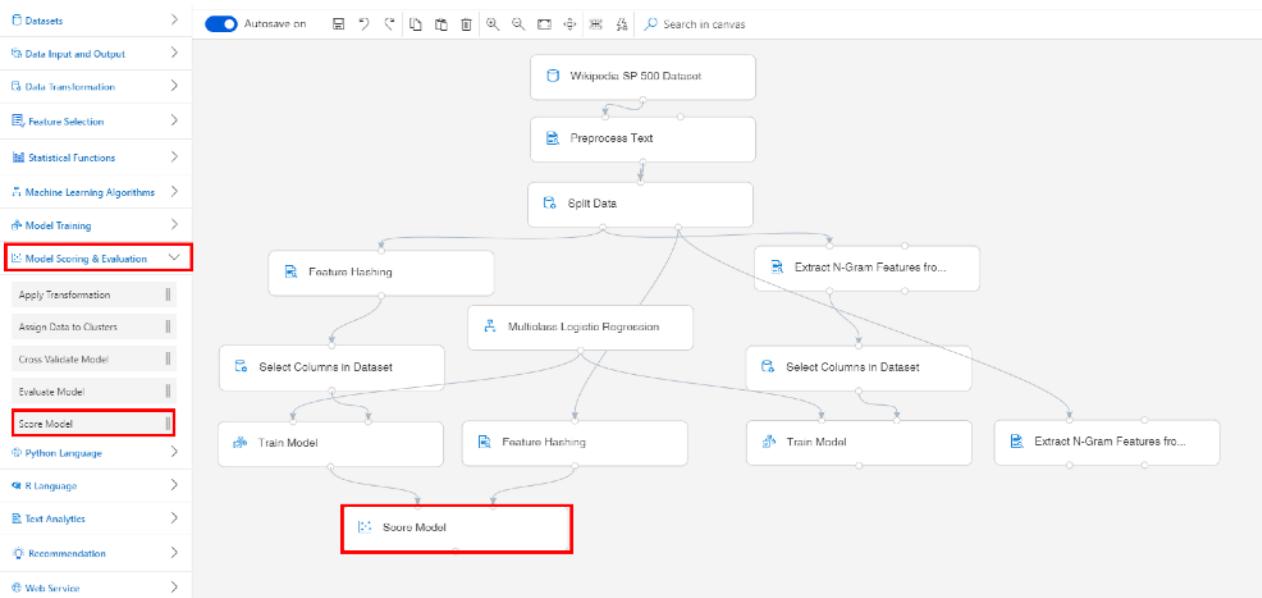
Task 12: Featurize unstructured text data with Extract N-Gram Feature from Text module, on the test set

1. Select the existing **Extract N-Gram Feature from Text** module.
2. Copy the **Extract N-Gram Feature from Text** module using the **Copy** option from the top menu.
3. **Paste** the copied module on the canvas.
4. Position the new **Extract N-Gram Feature from Text** module near the rightmost **Train Model** module on the canvas.
5. Connect the second output of the **Split Data**, the test set output, to the first input of the newly added **Extract N-Gram Feature from Text** module.
6. Connect the second output of the uppermost **Extract N-Gram Feature from Text**, to the second input of the copied **Extract N-Gram Feature from Text** module.
7. Select the newly added module and in the right settings pane, set the **Vocabulary mode** to **ReadOnly**

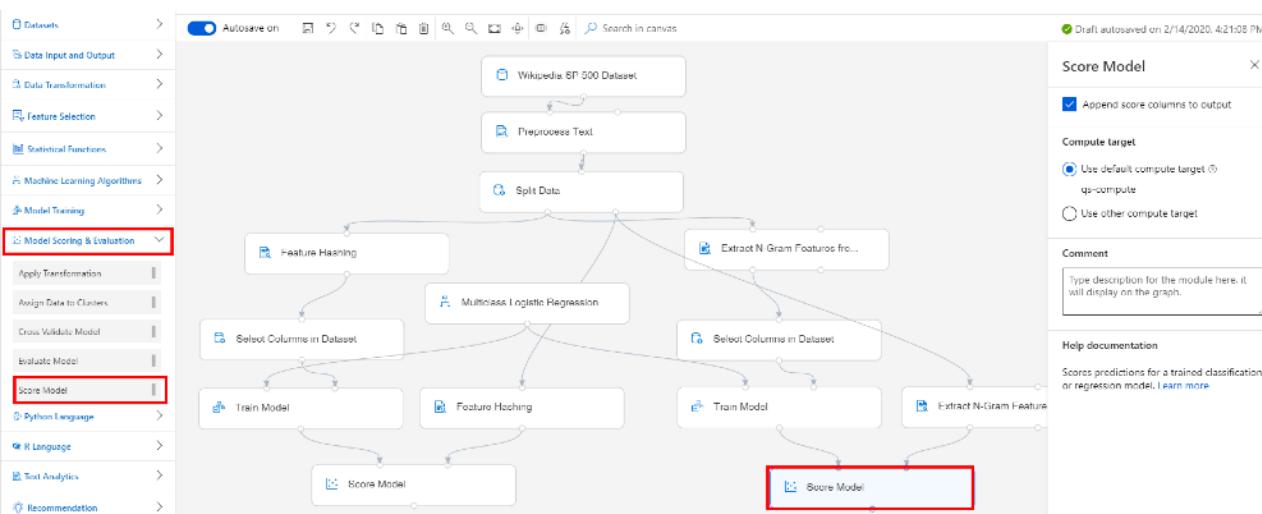


Task 13: Setup Score Model Modules

1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the leftmost **Train Model** module to the first input of the **Score Model** module
 4. Connect the output of the **Feature Hashing** module (the lower one, on the test set branch) to the second input of the **Score Model** module

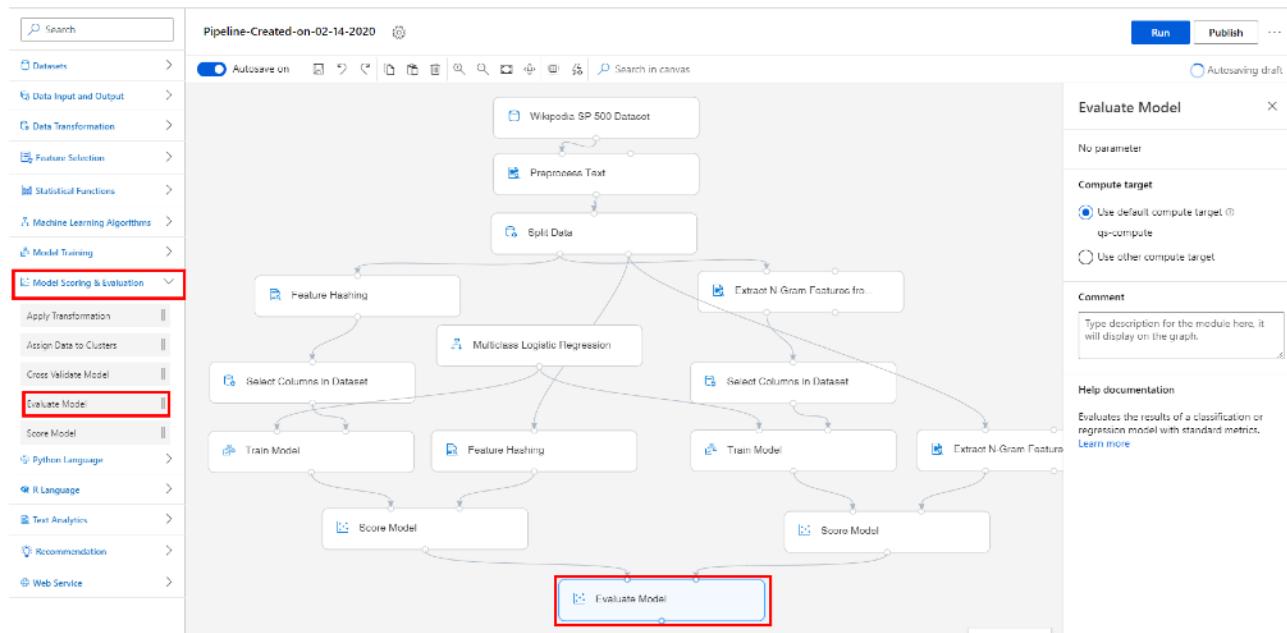


2. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Score Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the rightmost **Train Model** module to the first input of the **Score Model** module
 4. Connect the first output of the **Extract N-Gram Feature from Text** module (the lower one, on the test set branch) to the second input of the **Score Model** module



Task 14: Setup Evaluate Model Module

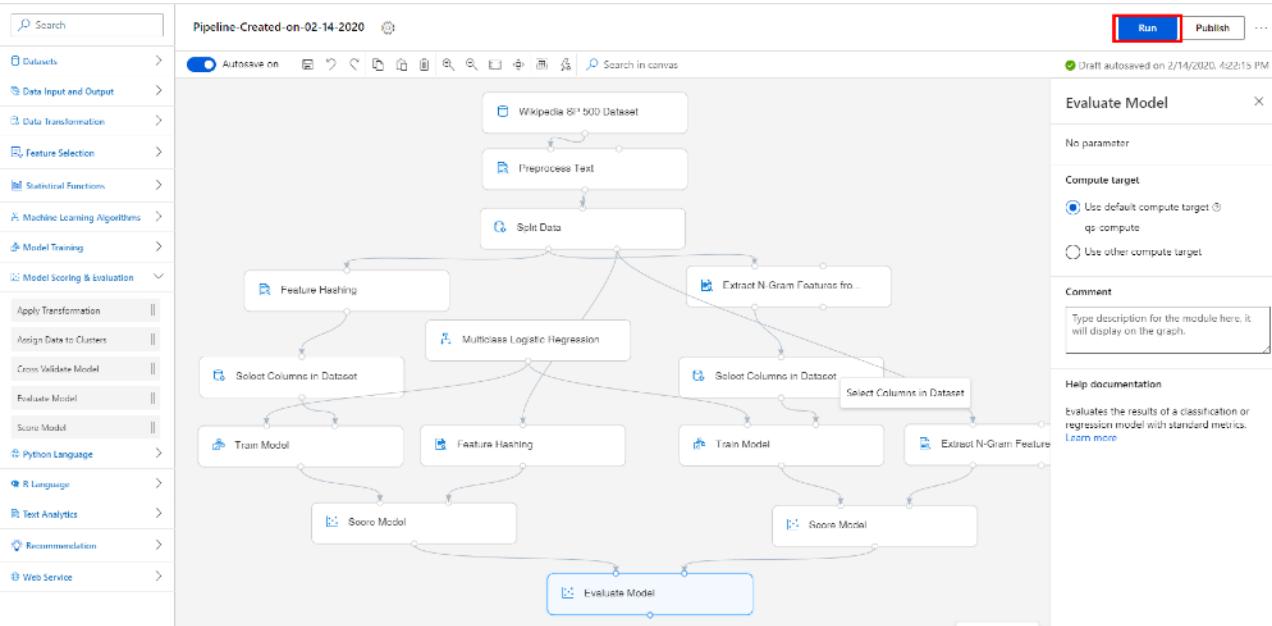
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
 1. Select the **Evaluate Model** prebuilt module
 2. Drag and drop the selected module on to the canvas
 3. Connect the two **Score Model** modules to the inputs of the **Evaluate Model** module



Exercise 2: Submit Training Pipeline

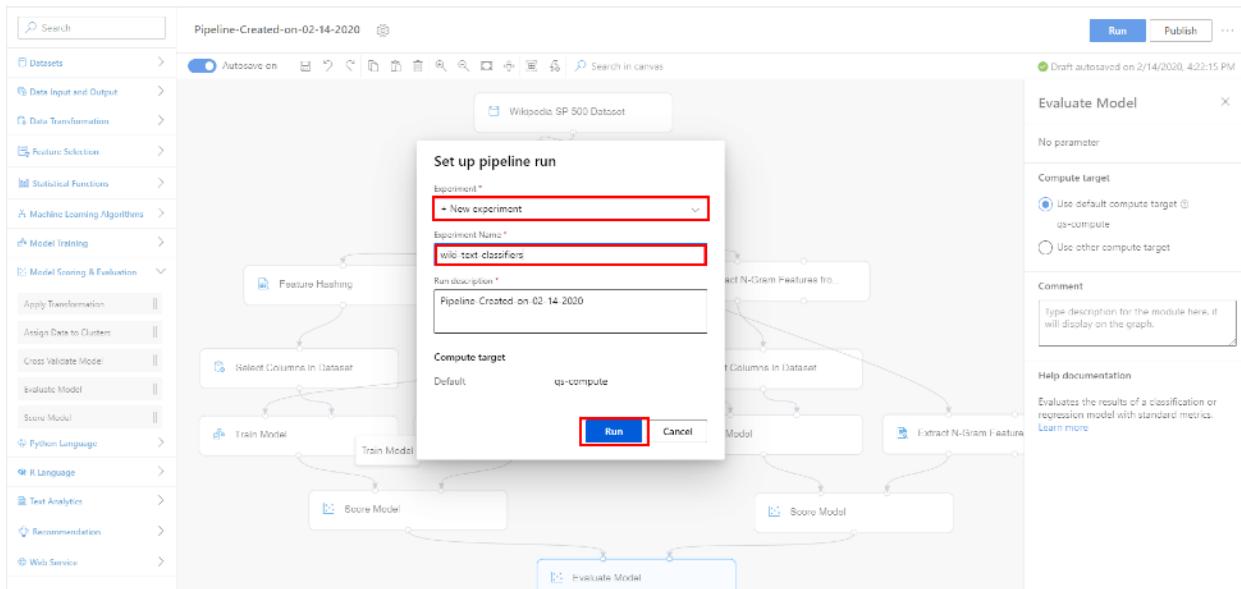
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the **Setup pipeline run editor**.



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

2. In the **Setup pipeline run editor**, select **Experiment, Create new** and provide **New experiment name: wiki-text-classifier**, and then select **Submit**.

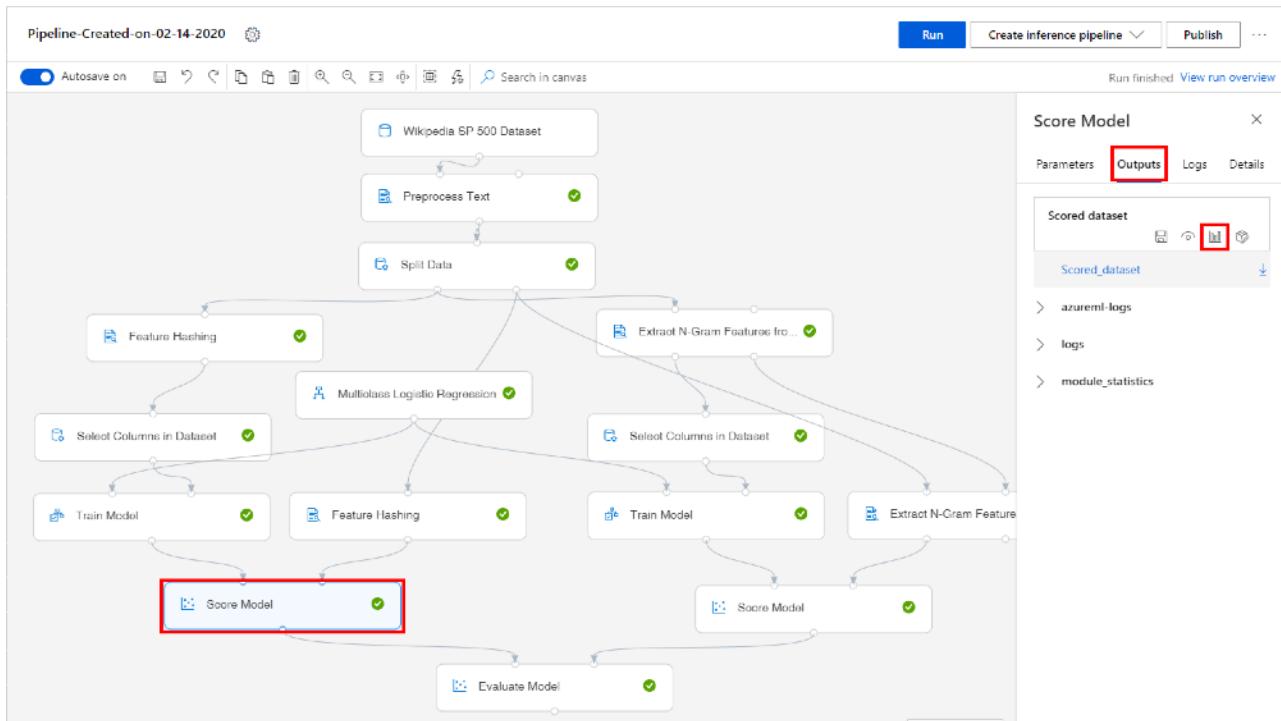


3. Wait for pipeline run to complete. It will take around **20 minutes** to complete the run.
4. While you wait for the model training to complete, you can learn more about the classification algorithm used in this lab by selecting **Multiclass Logistic Regression module**.

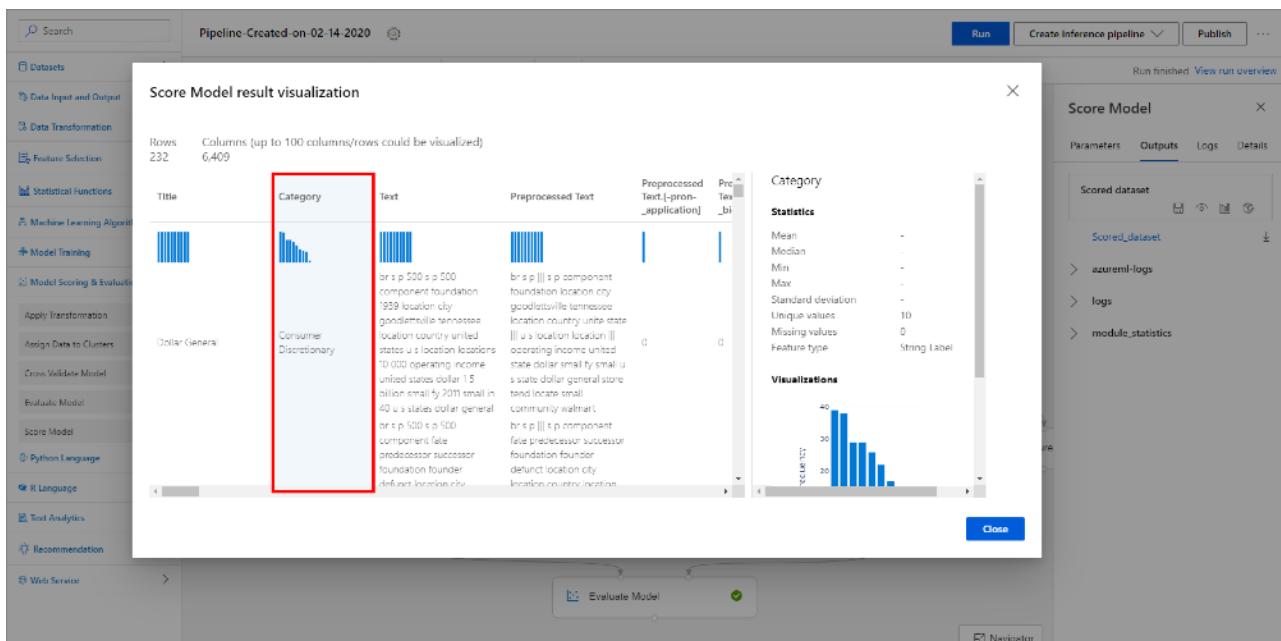
Exercise 3: Visualize Training Results

Task 1: Visualize the Model Predictions

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

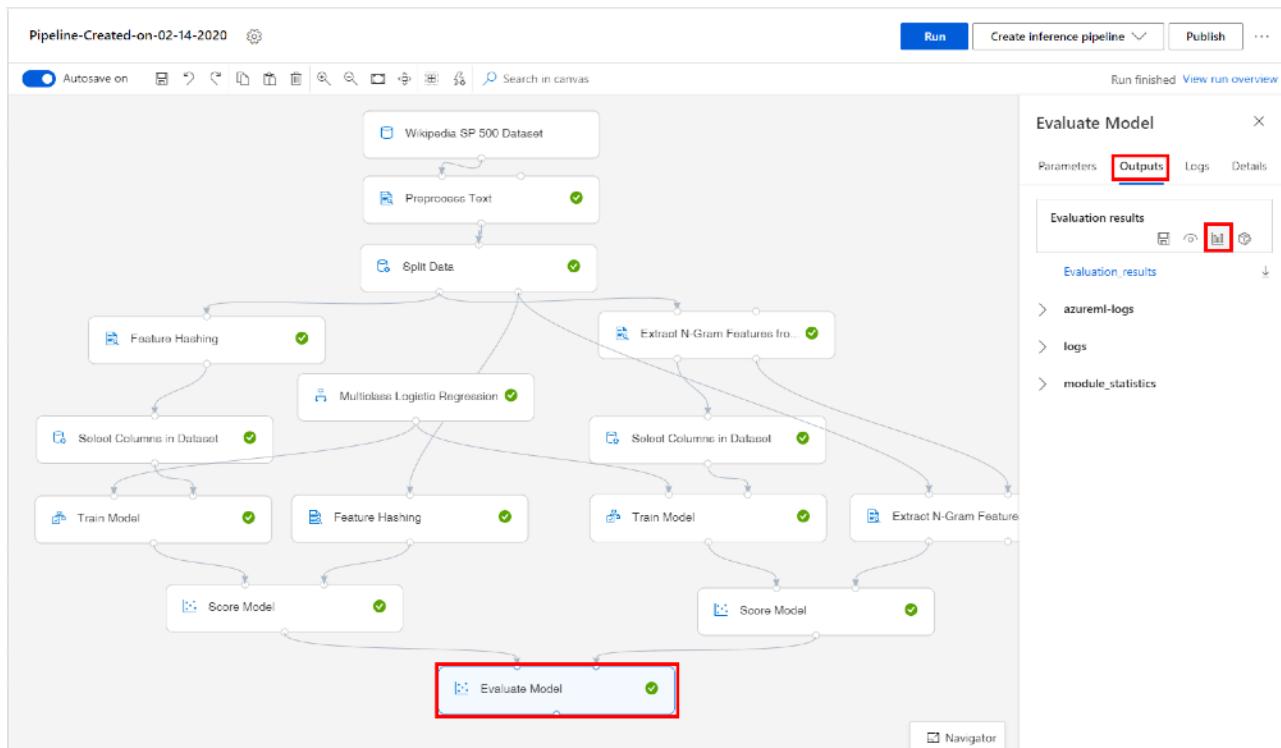


2. Observe the predicted values under the column **Category**.



Task 2: Visualize the Evaluation Results

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



2. Evaluate the model performance by reviewing the various evaluation metrics. Evaluate Model has two input ports, so that we could evaluate and compare scored datasets that are generated with different methods. In this sample, we compare the performance of the result generated with feature hashing method and n-gram method.



Next Steps

Congratulations! You have trained a simple text classifier and compared performance of the result generated with two different featurization modules. 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.

Lab 17 - Train a time-series forecasting model using Automated Machine Learning

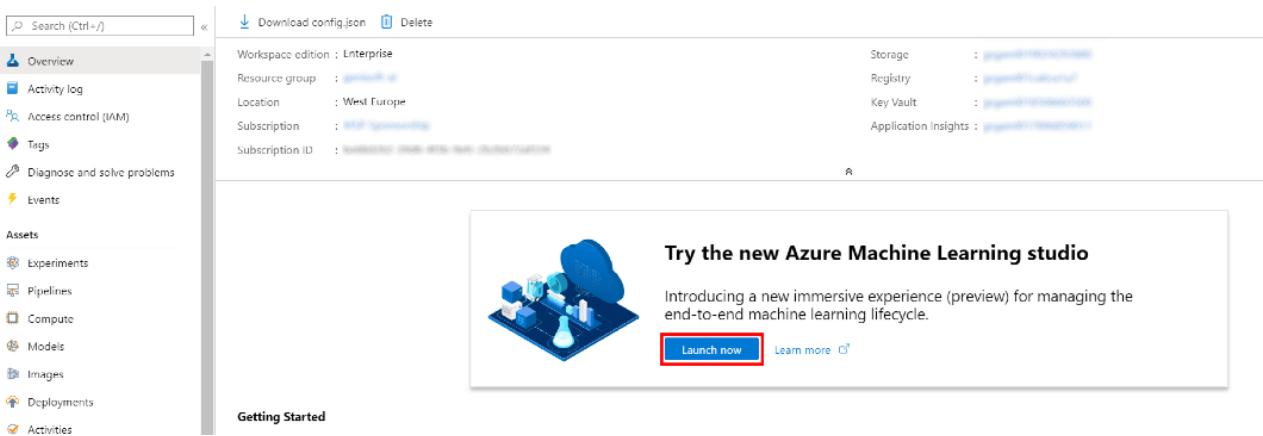
Overview

In this lab you will learn how the Automated Machine Learning capability in Azure Machine Learning (AML) can be used for the life cycle management of the manufactured vehicles and how AML helps in creation of better vehicle maintenance plans. To accomplish this, you will train a Linear Regression model to predict the number of days until battery failure using Automated Machine Learning available in AML studio.

Exercise 1: Creating a model using automated machine learning

Task 1: Create an automated machine learning experiment using the Portal

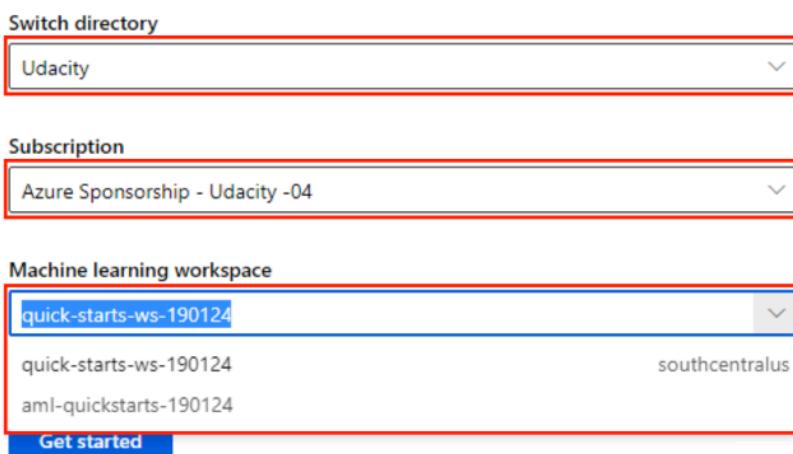
1. In [Azure portal](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



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**.

4. Select **Automated ML** in the left navigation bar.

The screenshot shows the Azure Machine Learning studio interface. The left sidebar has a red box around the 'Automated ML' option. The main content area is titled 'Welcome to the studio!' and contains four cards:

- Create new** (Notebooks): Code with Python SDK and run sample experiments. **Notebooks** icon. **Start now** button.
- Automated ML**: Automatically train and tune a model using a target metric. **Automated ML** icon. **Start now** button.
- Designer**: Drag-and-drop interface from prepping data to deploying models. **Designer** icon. **Start now** button.

5. Select **New automated ML run** to start creating a new experiment.

The screenshot shows the 'Automated ML' page. The left sidebar has a red box around the 'Automated ML' option. The main content area is titled 'Automated ML' and contains the following text:
Let Automated ML train and find the best model based on your data without writing a single line of code. [Learn more about Automated ML](#).
A red box surrounds the '+ New Automated ML run' button.
Text on the right: No recent Automated ML runs to display. Click "New Automated ML run" to create your first run.
[Learn more on creating Automated ML runs](#)

6. Select **Create dataset** and choose the **From web files** option from the drop-down.

The screenshot shows the 'Create a new Automated ML run' page. The left sidebar has a red box around the 'Select dataset' option. The main content area is titled 'Select dataset' and contains the following text:
Select a dataset from the list below, or create a new dataset. Automated ML currently only supports tabular data for authoring runs.
A red box surrounds the 'Create dataset' dropdown menu.
Options listed: From local files, From datastore, **From web files**, From Open Datasets.
Text on the right: Dataset type Tabular
Buttons: < Prev, Next >

7. Fill in the training data URL in the **Web URL** field: <https://introtomlsampleddata.blob.core.windows.net/data/battery-lifetime/training-formatted.csv>, make sure the name is set to **training-formatted-dataset**, and select **Next** to load a preview of the parsed training data.

Create dataset from web files

Basic info

Web URL *
https://introtomlsampleddata.blob.core.windows.net/data/battery-lifetime/training-formatted.csv

Name *
training-formatted-dataset

Dataset type * ⓘ
Tabular

Description
Dataset description

8. In the **Settings and preview** page, for the **Column headers** field, select **All files have same headers**. Scroll to the right to observe all of the columns in the data.

Create dataset from web files

Settings and preview

File format
Delimited

Delimiter
Comma Example: Field1,Field2,Field3

Encoding
UTF-8

Column headers
All files have same headers

Skip rows
None

Survival_In_Days	Province	Region	Trip_Length_Mean	Trip_Length_Sigma	Trips_Per_Day_Mean	Trips_Per_Day_Sigma
1283	Bretagne	West	18.10	6.03	4.73	1.18
1427	Occitanie	South	14.64	4.68	4.33	1.08
1436	Auvergne...	South	14.51	4.84	4.42	1.10
894	Martinique	West	20.85	6.95	4.28	1.07
1539	Reunion	South	11.58	3.96	4.56	1.14
1872	Marseille	South	14.07	4.69	4.70	1.17
151	Ile_de_Fra...	MidWest	13.39	4.46	4.54	1.13
1975	Normandie	MidWest	16.72	5.57	4.64	1.16
1957	Paris	MidWest	12.28	4.09	4.42	1.10
1150	Corse	South	19.62	6.54	4.32	1.08

Back **Next** **Cancel**

9. Select **Next** to check the schema and then confirm the dataset details by selecting **Next** and then **Create** on the confirmation page.

Create dataset from web files

Include	Column name	Properties	Type	Format settings and example
<input checked="" type="checkbox"/>	Path	Not applicable to selected type	String	
<input checked="" type="checkbox"/>	Survival_In_Days	Not applicable to selected type	Integer	1283, 1427, 1436
<input checked="" type="checkbox"/>	Province	Not applicable to selected type	String	Bretagne, Occitanie, Auvergne_Rhone_Alpes
<input checked="" type="checkbox"/>	Region	Not applicable to selected type	String	West, South, South
<input checked="" type="checkbox"/>	Trip_Length_Mean	Not applicable to selected type	Decimal	18.10325, 14.63707, 14.50564
<input checked="" type="checkbox"/>	Trip_Length_Sigma	Not applicable to selected type	Decimal	6.034416, 4.879023, 4.835215
<input checked="" type="checkbox"/>	Trips_Per_Day_Mean	Not applicable to selected type	Decimal	4.733162, 4.32595, 4.418737
<input checked="" type="checkbox"/>	Trips_Per_Day_Sigma	Not applicable to selected type	Decimal	1.163291, 1.081487, 1.104684
<input checked="" type="checkbox"/>	Battery_Rated_Cycles	Not applicable to selected type	Integer	275, 250, 250
<input checked="" type="checkbox"/>	Manufacture_Month	Not applicable to selected type	String	M8, M8, M9
<input checked="" type="checkbox"/>	Manufacture_Year	Not applicable to selected type	String	Y2010, Y2014, Y2018
<input checked="" type="checkbox"/>	Alternator_Efficiency	Not applicable to selected type	Decimal	0.9812054, 0.9224901, 0.990107
<input checked="" type="checkbox"/>	Car_Has_EcoStart	Not applicable to selected type	Boolean	false, true, false
<input checked="" type="checkbox"/>	Twelve_hourly_temperature_hist...	Not applicable to selected type	Decimal	15.47809, 15.30073, 18.23024

Back **Next** **Cancel**

10. Now you should be able to select the newly created dataset for your experiment. Select the **training-formatted-dataset** dataset and select **Next** to move to the experiment run details page.

quick-starts-ws-148831 > Automated ML > Start run

Success : training-formatted-dataset dataset created successfully

Create a new Automated ML run

Select dataset

Select dataset

Configure run

Task type and settings

Select a dataset from the list below, or create a new dataset. Automated ML currently only supports tabular data for authoring runs.

+ Create dataset **Show supported datasets only**

Dataset name	Dataset type
<input checked="" type="radio"/> training-formatted-dataset	Tabular

11. You will now configure the Auto ML run basic settings by providing the following values for the experiment name, target column and training compute:
- Experiment name: **automlregression**
 - Target column: select **Survival_In_Days**
 - Select training compute target: : select **qs-compute**

Success: training-formatted-dataset dataset created successfully

Create a new Automated ML run

Configure run

Configure the experiment. Select from existing experiments or define a new name, select the target column and the training compute to use. [Learn more on how to configure the experiment](#)

Dataset
training-formatted-dataset ([View dataset](#))

Experiment name *

Target column *

Select training compute target *

[Create a new compute](#) [Refresh compute](#)

12. Select **Next** and select **Regression** in the **Task type and settings** page.

Create a new Automated ML run

Select task type

Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.

Classification
To predict one of several categories in the target column: yes/no, blue, red, green.

Regression
To predict continuous numeric values

Time series forecasting
To predict values based on time

[View additional configuration settings](#) [View featurization settings](#)

13. Select **View additional configuration settings** to open the advanced settings section. Provide the following settings:

- Primary metric: **Normalized root mean squared error**
- Exit criterion > Metric score threshold: **0.09**
- Validation > Validation type: **k-fold cross validation**
- Validation > Number of Cross Validations: **5**
- Concurrency > Max concurrent iterations: **1**

Additional configurations

Primary metric ⓘ
Normalized root mean squared error

Automatic featurization ⓘ

Explain best model ⓘ

Blocked algorithms ⓘ
A list of algorithms that automated ML will not use during training.

Exit criterion
Training job time (hours) ⓘ 3

Metric score threshold ⓘ 0.09

Validation
Validation type ⓘ k-fold cross validation

Number of Cross Validations * ⓘ 5

Concurrency
Max concurrent iterations ⓘ 1

Save

14. Select **Save** and then **Finish** to begin the automated machine learning process.

Create a new automated machine learning run

Select task type

Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.

Classification
To predict one of several categories in the target column, yes/no, blue, red, green.

Regression
To predict continuous numeric values

Creating a new automated machine learning run...

Validating data...

15. Wait until the Run status becomes **Running** in the Run Detail page.

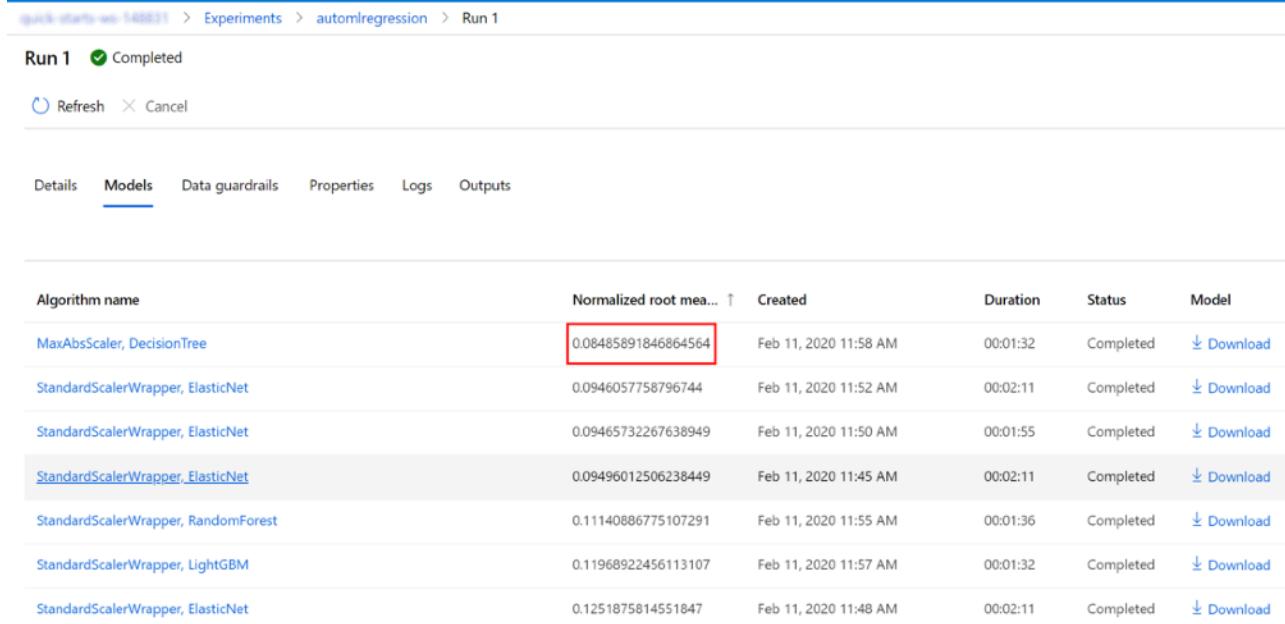
The screenshot shows the 'Run details' section of the Run Detail page. At the top, it says 'Run 1' with a green checkmark icon and the status 'Running'. Below that are buttons for 'Refresh' and 'Cancel'. The main area has tabs for 'Details', 'Models', 'Data guardrails', 'Properties', 'Logs', and 'Outputs', with 'Details' being the active tab. Under 'Run details', it shows 'Task type: regression', 'Primary metric: Normalized root mean squared error', and 'Run status: Running'. At the bottom, it shows 'Experiment name: automl-regression' and 'Run ID: AutoML_d13f8408-9d02-496b-937b-ed9fe2f20dd8'.

Task 2: Review the experiment run results

1. The experiment will run for about 15 minutes. While it runs and once it completes, you should check the Models tab on the Run Detail page to observe the model performance for the primary metric for different runs.

The screenshot shows the 'Models' tab selected on the Run Detail page. At the top, it says 'Run 1' with a green checkmark icon and the status 'Completed'. Below that are buttons for 'Refresh' and 'Cancel'. The main area has tabs for 'Details', 'Models', 'Data guardrails', 'Properties', 'Logs', and 'Outputs', with 'Models' being the active tab. Under 'Recommended model', it shows 'Model name: MaxAbsScaler, DecisionTree', 'Metric value: 0.08485891846864564', 'Started on: Feb 11, 2020 11:59 AM', 'Duration: 00:01:32', 'Sdk version: 1.1.0rc0', and 'Deploy status: No deployment yet'. To the right, under 'Run summary', it shows 'Task type: Regression', 'Primary metric: Normalized root mean squared error', 'Run status: Completed', 'Experiment name: automlregression', and 'Run ID: AutoML_14918412-5538-402b-b7dd-008e17fd2402'. At the bottom, there are buttons for 'Deploy best model', 'View model details' (which is highlighted with a red box), and 'Download best model'.

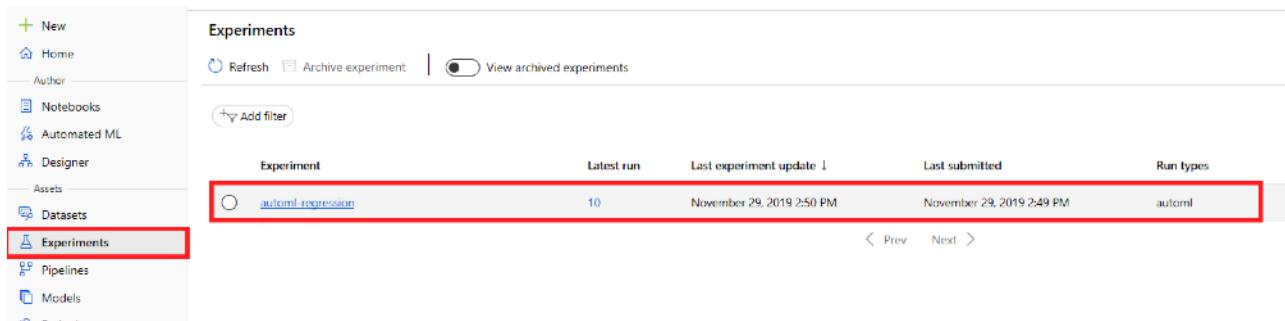
2. In the models list, notice at the top the iteration with the best **normalized root mean square error** score. Note that the normalized root mean square error measures the error between the predicted value and actual value. In this case, the model with the lowest normalized root mean square error is the best model.



The screenshot shows the Azure Machine Learning studio interface. The URL is [quick-starts-ws-148831 / Experiments / automlregression / Run 1](#). The page title is "Run 1 Completed". Below it are "Refresh" and "Cancel" buttons. A navigation bar includes "Details", "Models" (which is underlined), "Data guardrails", "Properties", "Logs", and "Outputs". The main content is a table titled "Algorithm name" with columns: "Normalized root mea...", "Created", "Duration", "Status", and "Model". The table lists several models with their scores, creation times, durations, and download links. The first row, "MaxAbsScaler, DecisionTree", has a red box around its score: 0.08485891846864564.

Algorithm name	Normalized root mea...	Created	Duration	Status	Model
MaxAbsScaler, DecisionTree	0.08485891846864564	Feb 11, 2020 11:58 AM	00:01:32	Completed	Download
StandardScalerWrapper, ElasticNet	0.0946057758796744	Feb 11, 2020 11:52 AM	00:02:11	Completed	Download
StandardScalerWrapper, ElasticNet	0.09465732267638949	Feb 11, 2020 11:50 AM	00:01:55	Completed	Download
StandardScalerWrapper, ElasticNet	0.09496012506238449	Feb 11, 2020 11:45 AM	00:02:11	Completed	Download
StandardScalerWrapper, RandomForest	0.11140886775107291	Feb 11, 2020 11:55 AM	00:01:36	Completed	Download
StandardScalerWrapper, LightGBM	0.11968922456113107	Feb 11, 2020 11:57 AM	00:01:32	Completed	Download
StandardScalerWrapper, ElasticNet	0.1251875814551847	Feb 11, 2020 11:48 AM	00:02:11	Completed	Download

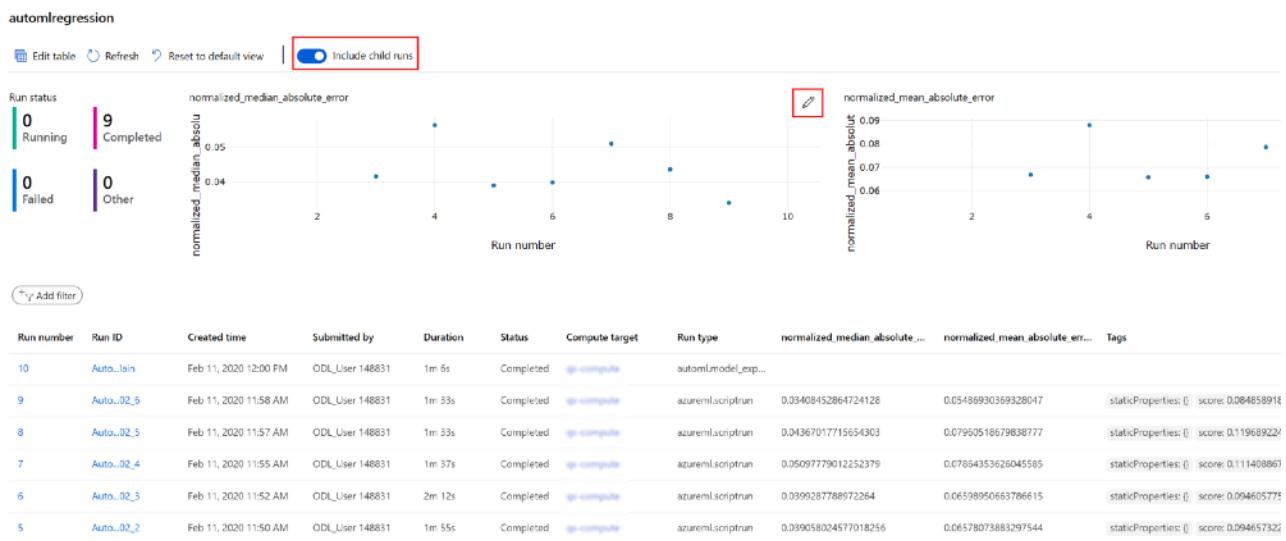
3. Select **Experiments** on the left navigation pane and select the experiment **automlregression** to see the list of available runs.



The screenshot shows the Azure Machine Learning studio interface. The left sidebar has a red box around the "Experiments" item under the "Datasets" section. The main area is titled "Experiments" with "10" runs listed. The table columns are "Experiment", "Latest run", "Last experiment update", "Last submitted", and "Run types". The first row, "automlregression", has a red box around it. Navigation buttons "Prev" and "Next" are at the bottom.

Experiment	Latest run	Last experiment update	Last submitted	Run types
automlregression	10	November 29, 2019 2:50 PM	November 29, 2019 2:49 PM	automl

4. Select the option to **Include child runs** to be able to examine model performance for the primary metric of different runs. By default, the left chart describes the `normalized_median_absolute_error` value for each run. Select the pen icon on the right corner of the `normalized_median_absolute_error` chart to configure the `normalized_root_mean_square_error` metric representation.



Next Steps

Congratulations! You have trained a simple time-series forecasting model using automated machine learning in the visual interface. 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.

Lab 18 - Managing a compute instance

Machine learning requires several tools to prepare data, and train and deploy models. Most of the work usually takes place within web-based, interactive notebooks, such as Jupyter notebooks. Although notebooks are lightweight and easily run in a web browser, you still need a server to host them.

So, the setup process for most users is to install several applications and libraries on a machine, configure the environment settings, then load any additional resources to begin working within notebooks or integrated development environments (IDEs). All this setup takes time, and there is sometimes a fair amount of troubleshooting involved to make sure you have the right combination of software versions that are compatible with one another.

What if you could use a ready-made environment that is pre-optimized for your machine learning development?

Azure Machine Learning [compute instance](#) provides this type of environment for you, and is fully managed, meaning you don't have to worry about setup and applying patches and updates to the underlying virtual machine. Plus, since it is cloud-based, you can run it from anywhere and from any machine. All you need to do is specify the type of virtual machine, including GPUs and I/O-optimized options, then you have what you need to start working.

The managed services, such as computer instance and compute cluster, can be used as a training compute target to scale out training resources to handle larger data sets. When you are ready to run your experiments and build your models, you need to specify a compute target. Compute targets are compute resources where you run your experiments or host your service deployment. The target may be your local machine or a cloud-based resource. This is another example of where managed services like compute instance and computer cluster really shine.

A managed compute resource is created and managed by Azure Machine Learning. This compute is optimized for machine learning workloads.

Azure Machine Learning compute clusters and compute instances are the only managed computes. Additional managed compute resources may be added in the future.

Overview

In this lab, you will explore different actions you can take to manage a compute instance in Azure Machine Learning Studio.

Exercise 1: Create New Compute Instance

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

The screenshot shows the Azure Machine Learning workspace overview page. On the left, there's a sidebar with various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets, Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The main area displays workspace details: Edition (Enterprise), Resource group (ml-quickstarts), Location (West Europe), Subscription (ml-quickstarts), and Subscription ID (ml-quickstarts). Below this, there's a 'Getting Started' section with a card titled 'Try the new Azure Machine Learning studio'. The card contains a small 3D model of a city on a circuit board, the text 'Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle.', and two buttons: 'Launch now' (which is highlighted with a red box) and 'Learn more'.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

The screenshot shows the 'Welcome to the studio!' setup screen. It has three dropdown menus with red boxes around them:

- Switch directory:** Set to 'Udacity'.
- Subscription:** Set to 'Azure Sponsorship - Udacity -04'.
- Machine learning workspace:** Shows a list of workspaces:
 - 'quick-starts-ws-190124' (selected, highlighted with a blue border)
 - 'quick-starts-ws-190124'
 - 'aml-quickstarts-190124'

A blue 'Get started' button is located at the bottom of the 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**.

4. From the studio, navigate to **Compute**, then select **+New**.

5. In the **New Compute Instance** pane, provide the following information and then select **Create**.
- Compute name: **provide an unique name**
 - Virtual Machine size: **Standard_D3_v2**

6. It will take couple of minutes for your compute instance to be ready. Wait for your compute instance to be in status **Running**.

Exercise 2: Explore Compute Instances

1. Select the radio button next to the name of your compute instance. This will select the instance, as indicated by a checkmark. Selecting your instance in this way enables the toolbar options above that enable you to Stop, Restart, or Delete the instance.

Compute

The screenshot shows the 'Compute Instances' blade in the Azure portal. At the top, there are tabs for 'Compute Instances', 'Training clusters', 'Inference clusters', and 'Attached compute'. Below the tabs, a message states: '(i) Compute instance is replacing the Notebook VM. You cannot create new Notebook VMs, but you can still use existing N...'. A toolbar at the top right includes 'New', 'Refresh', 'Start', 'Stop' (radio button selected), 'Restart', 'Delete', and a toggle switch for 'Show created by me only'. A red box highlights the 'Stop', 'Restart', and 'Delete' buttons. Another red box highlights the checkmark in the 'Name' column for the 'compute-instances' row. The table below has columns for 'Name', 'Status', and 'Application URI'. The 'compute-instances' row shows 'Running' status and links for 'JupyterLab', 'Jupyter', 'RStudio', and 'SSH'. Navigation arrows for 'Prev' and 'Next' are at the bottom.

Name	Status	Application URI
compute-instances	Running	JupyterLab Jupyter RStudio SSH

There are different scenarios in which you will want to perform these actions. Here are the actions you can take on a selected compute instance, and what they do:

- **Stop:** Since the compute instance runs on a virtual machine (VM), you pay for the instance as long as it is running. Naturally, it needs to run to perform compute tasks, but when you are done using it, be sure to stop it with this option to prevent unnecessary costs.
- **Restart:** Restarting an instance is sometimes necessary after installing certain libraries or extensions. There may be times, however, when the compute instance stops functioning as expected. When this happens, try restarting it before taking further action.
- **Delete:** You can create and delete instances as you see fit. The good news is, all notebooks and R scripts are stored in the default storage account of your workspace in Azure file share, within the "User files" directory. This central storage allows all compute instances in the same workspace to access the same files so you don't lose them when you delete an instance you no longer need.

-
2. Select the name of your instance. This opens the **Compute details** blade, revealing useful information about your compute instance.

Compute details

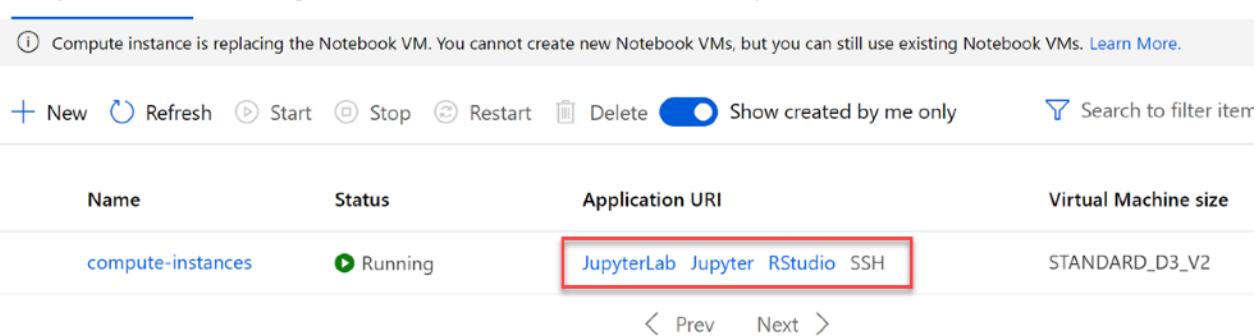
Refresh Delete

Attributes	Resource properties
Compute name compute-instances	Status Running
Compute type Compute Instance	Virtual Machine size STANDARD_D3_V2
Subscription ID [REDACTED]	Application URI JupyterLab Jupyter RStudio SSH
Resource group intro-to-ml	Created on [REDACTED]
Workspace intro-to-ml-workspace	SSH access Disabled
Region centralus	Private IP address 10.0.0.4
	Public IP address 52.141.222.189
	Virtual network/subnet --

3. The **Attributes** describe the resource details of the compute instance, including the name, type, Azure subscription, the resource group to which it belongs, the Azure Machine Learning workspace that manages it, and the Azure region to which it is deployed. If you need to execute scripts that require details about your compute instance, this is where you can find most of what you need. The **Resource properties** show the status and configuration of the compute instance, including links to its applications and public and private endpoints. In this screenshot, you will see that SSH access is disabled. You cannot enable SSH access after creating a compute instance. You can only enable this option at the time of creation. SSH access allows you to securely connect to the VM from a terminal or command window. Use the public IP address to connect via SSH or an integrated development environment (IDE) like [Visual Studio Code](#).

-
4. Navigate back to **Compute**. The compute instance comes preconfigured with tools and environments that enable you to author, train, and deploy models in a fully integrated notebook experience. You access these environments through the **Application URI** links located in the resource properties (as seen in the previous step), and next to each compute instance in the list.

Compute



The screenshot shows the 'Compute Instances' page with the following details:

Name	Status	Application URI	Virtual Machine size
compute-instances	Running	JupyterLab Jupyter RStudio SSH	STANDARD_D3_V2

A red box highlights the 'Application URI' column for the 'compute-instances' row. The 'Application URI' column contains four links: JupyterLab, Jupyter, RStudio, and SSH. The 'Status' column shows 'Running'. The 'Name' column shows 'compute-instances'. The 'Virtual Machine size' column shows 'STANDARD_D3_V2'. Navigation buttons at the bottom include '< Prev' and 'Next >'.

5. Select each of the application links to sign in to the related environment. You may be prompted to select your user account for each application.

Next Steps

Congratulations! You have completed the introduction to managing a compute instance lab. 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.

Lab 19 - Train a machine learning model from a managed notebook environment

So far, the Managed Services for Azure Machine Learning lesson has covered **compute instance** and the benefits it provides through its fully managed environment containing everything you need to run Azure Machine Learning. Now it is time to gain some hands-on experience by putting a compute instance to work.

Overview

In this lab, you learn the foundational design patterns in Azure Machine Learning, and train a simple scikit-learn model based on the diabetes data set. After completing this lab, you will have the practical knowledge of the SDK to scale up to developing more-complex experiments and workflows.

In this tutorial, you learn the following tasks:

- Connect your workspace and create an experiment
- Load data and train a scikit-learn model

Exercise 1: Run the Notebook for this Lab

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

The screenshot shows the Azure Machine Learning workspace overview page. On the left, there's a sidebar with various navigation links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets, Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The main area displays workspace details: Edition (Enterprise), Resource group, Location (West Europe), Subscription (mlt-quickstarts), and Subscription ID. Below this, there's a 'Try the new Azure Machine Learning studio' section featuring a 3D icon of a city on a cloud, a brief introduction, and a prominent 'Launch now' button which is highlighted with a red box.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

The screenshot shows the 'Welcome to the studio!' setup screen. It has three main dropdown sections: 'Switch directory' (set to 'Udacity'), 'Subscription' (set to 'Azure Sponsorship - Udacity -04'), and 'Machine learning workspace' (set to 'quick-starts-ws-190124'). The 'Machine learning workspace' dropdown also lists 'quick-starts-ws-190124' and 'aml-quickstarts-190124'. A blue 'Get started' button is visible at the bottom of this section. All three dropdowns are highlighted with red boxes.

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**.

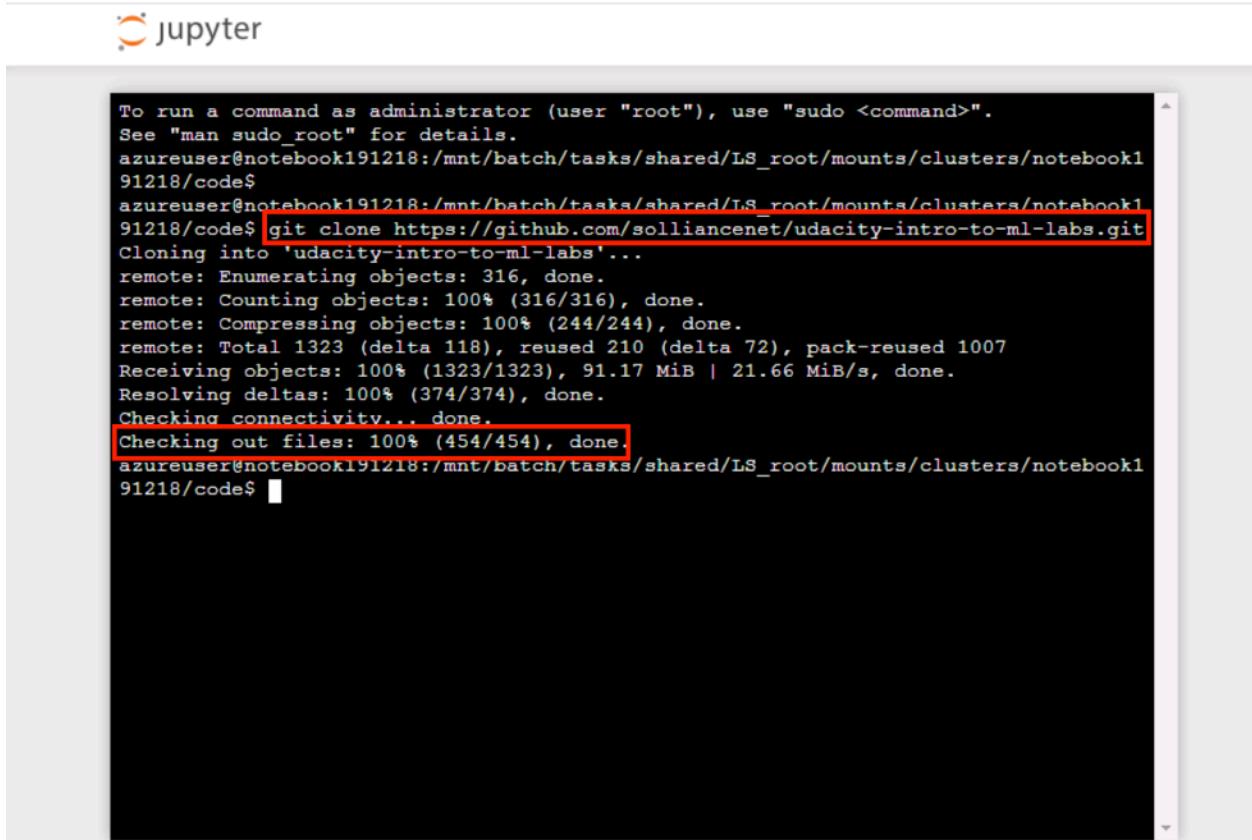
4. From the studio, navigate to **Compute**. Next, for the available Compute Instance, under Application URI select **Jupyter**. Be sure to select **Jupyter** and not **JupyterLab**.

The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a navigation sidebar with various options like New, Home, Author, Notebooks, Automated ML, Designer, Assets, Datasets, Experiments, Pipelines, Models, Endpoints, Manage, Compute (which is highlighted with a red box), Datastores, and Data Labeling. The main area is titled "Compute" and shows a list of "Compute Instances". The list includes columns for Name, Status, Application URI, Virtual Machine size, and Created on. There are buttons for New, Refresh, Start, Stop, Restart, Delete, Show created by me only, and Search to filter items. A note at the top says "Compute instance is replacing the Notebook VM. You cannot create new Notebook VMs, but you can still use existing Notebook VMs. Learn More." One instance is listed: Name is "quick-starts-ws-191218", Status is "Running", Application URI is "JupyterLab Jupyter RStudio SSH" (with "Jupyter" highlighted with a red box), and Virtual Machine size is "STANDARD_D3_V2".

5. From within the Jupyter interface, select **New, Terminal**.

The screenshot shows the Jupyter interface. At the top, there are two browser tabs: one for "manage.cloudlabs.ai" and another for "notebook191218.southcentralus.instances.azureml.net". The main window has a title bar "jupyter" and a menu bar with "Files", "Running", "Clusters", "AzureML Samples", "Upload", "New", and "Quit". A context menu is open, listing "Notebook:" with options "Python 3", "Python 3.6 - AzureML", and "R", and "Other:" with "Text File" and "Folder". The "Terminal" option is at the bottom of the list and is highlighted with a red box, with a cursor pointing at it.

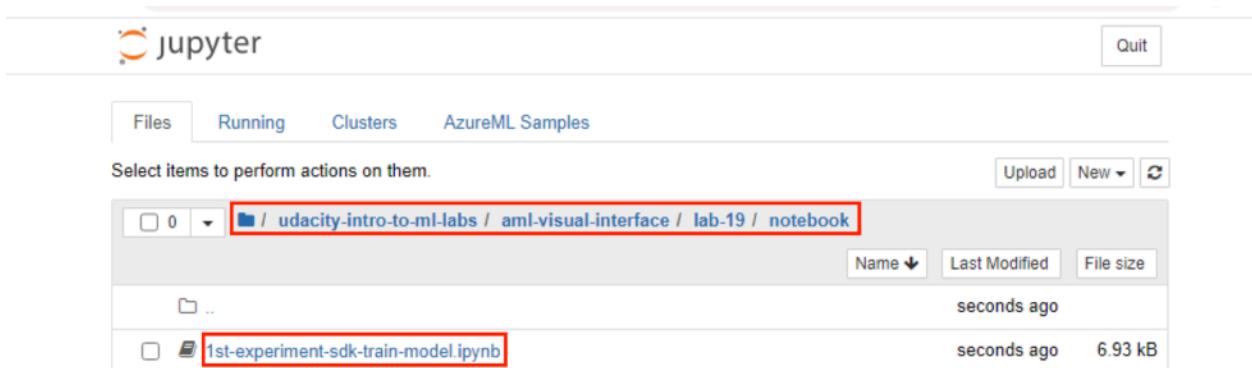
6. In the new terminal window run the following command and wait for it to finish:
`git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git`



The screenshot shows a terminal window titled "jupyter". The command `git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git` is highlighted with a red box. The terminal output shows the progress of cloning a repository from GitHub, including object enumeration, counting, compressing, receiving objects, and resolving deltas. The final message indicates successful connectivity and file checkout.

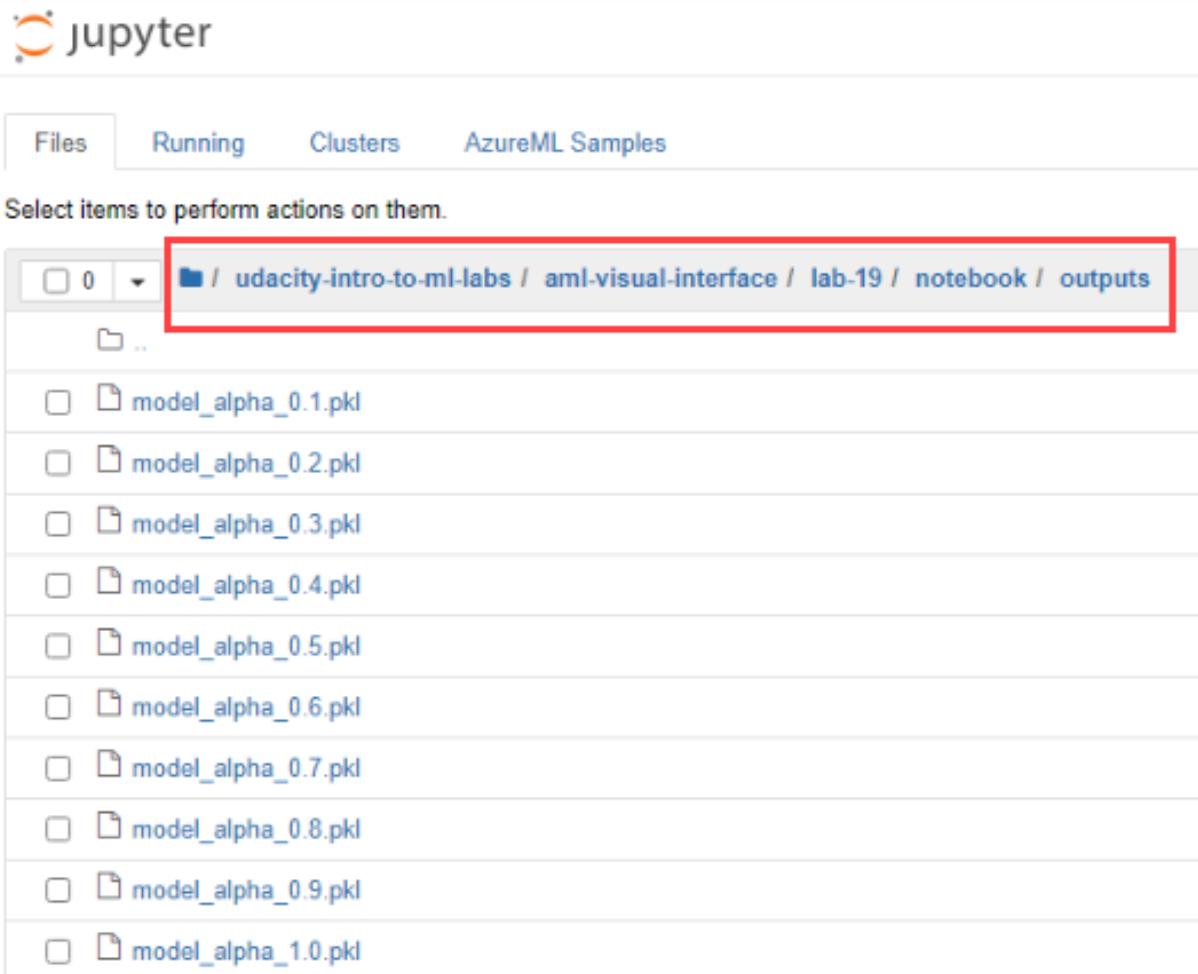
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$ git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git
Cloning into 'udacity-intro-to-ml-labs'...
remote: Enumerating objects: 316, done.
remote: Counting objects: 100% (316/316), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 1323 (delta 118), reused 210 (delta 72), pack-reused 1007
Receiving objects: 100% (1323/1323), 91.17 MiB | 21.66 MiB/s, done.
Resolving deltas: 100% (374/374), done.
Checking connectivity... done.
Checking out files: 100% (454/454), done.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$
```

7. From within the Jupyter interface, navigate to directory `udacity-intro-to-ml-labs/aml-visual-interface/lab-19/notebook` and open `1st-experiment-sdk-train-model.ipynb`. This is the Python notebook you will step through executing in this lab.



8. Follow the instructions within the notebook to complete the lab.

-
9. After completing the notebook, navigate back to the **Notebook** folder, then select the newly created **outputs** sub-folder. Here you will see the trained models (***.pkl** files) generated by the last cell you executed. In addition, the serialized model is uploaded to each run. This allows you to download the model file from the run in the portal as an alternative to downloading them from this folder.



Next Steps

Congratulations! You have just learned how to use the Jupyter application on a compute instance to train a model. You can now return to the Udacity portal to continue with the lesson.

Lab 20 - Explore experiments and runs

In the previous lab (19), you executed a Jupyter notebook that trained a model through a series of 10 different runs, each with a different alpha hyperparameter applied. These runs were created within the experiment you created at the beginning of the notebook. Because of this, Azure Machine Learning logged the details so you can review the result of each run and see how the alpha value is different between them.

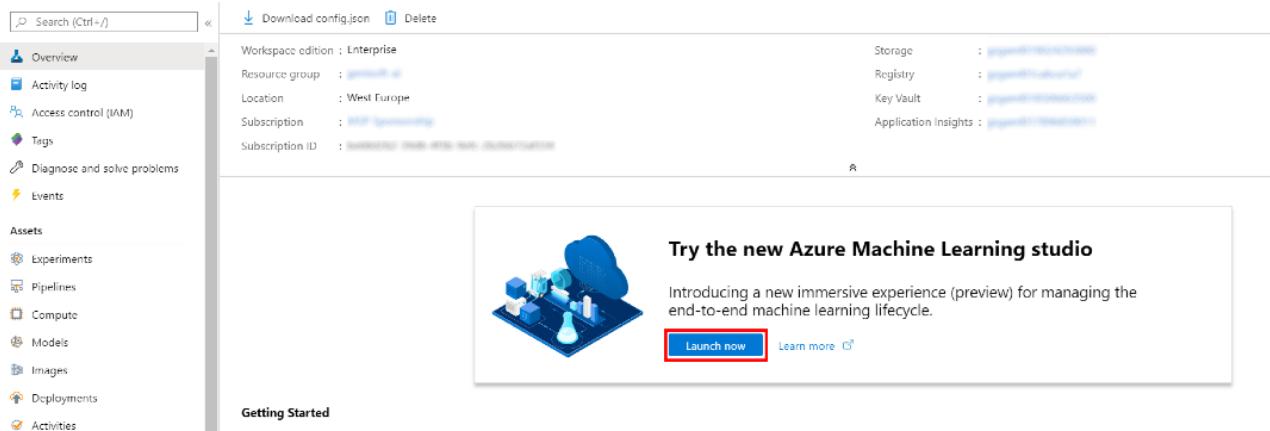
Overview

In this lab, you view the experiments and runs executed by a notebook. In the first part of the lab, you will use a notebook to create and run the experiments. In the second part of the lab, you will navigate to the **Experiments** blade in Azure Machine Learning Studio. Here you see all the individual runs in the experiment. Any custom-logged values (alpha_value and rmse, in this case) become fields for each run, and also become available for the charts and tiles at the top of the experiment page. To add a logged metric to a chart or tile, hover over it, click the edit button, and find your custom-logged metric.

When training models at scale over hundreds and thousands of separate runs, this page makes it easy to see every model you trained, specifically how they were trained, and how your unique metrics have changed over time.

Exercise 1: Run the Notebook for this Lab

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

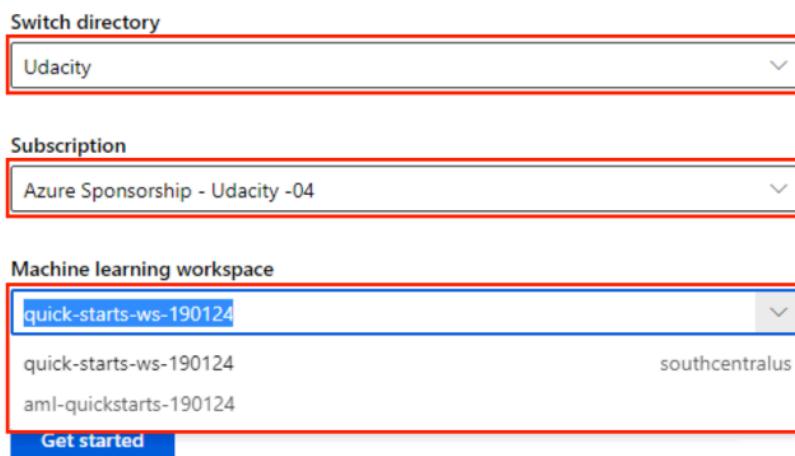


The screenshot shows the Azure Machine Learning workspace overview page. On the right, there's a promotional section for the new Azure Machine Learning studio, featuring a 3D icon of a lab setup and the text "Try the new Azure Machine Learning studio". Below it is a sub-section titled "Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle." with a "Launch now" button (which is highlighted with a red box) and a "Learn more" link. The left sidebar contains a navigation menu with items such as Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets, Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The "Getting Started" section is also visible.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



The screenshot shows the "Welcome to the studio!" configuration screen. It has three main dropdown menus: "Switch directory" (set to "Udacity"), "Subscription" (set to "Azure Sponsorship - Udacity -04"), and "Machine learning workspace" (listing "quick-starts-ws-190124", "quick-starts-ws-190124", and "aml-quickstarts-190124"). At the bottom is a blue "Get started" button.

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**.

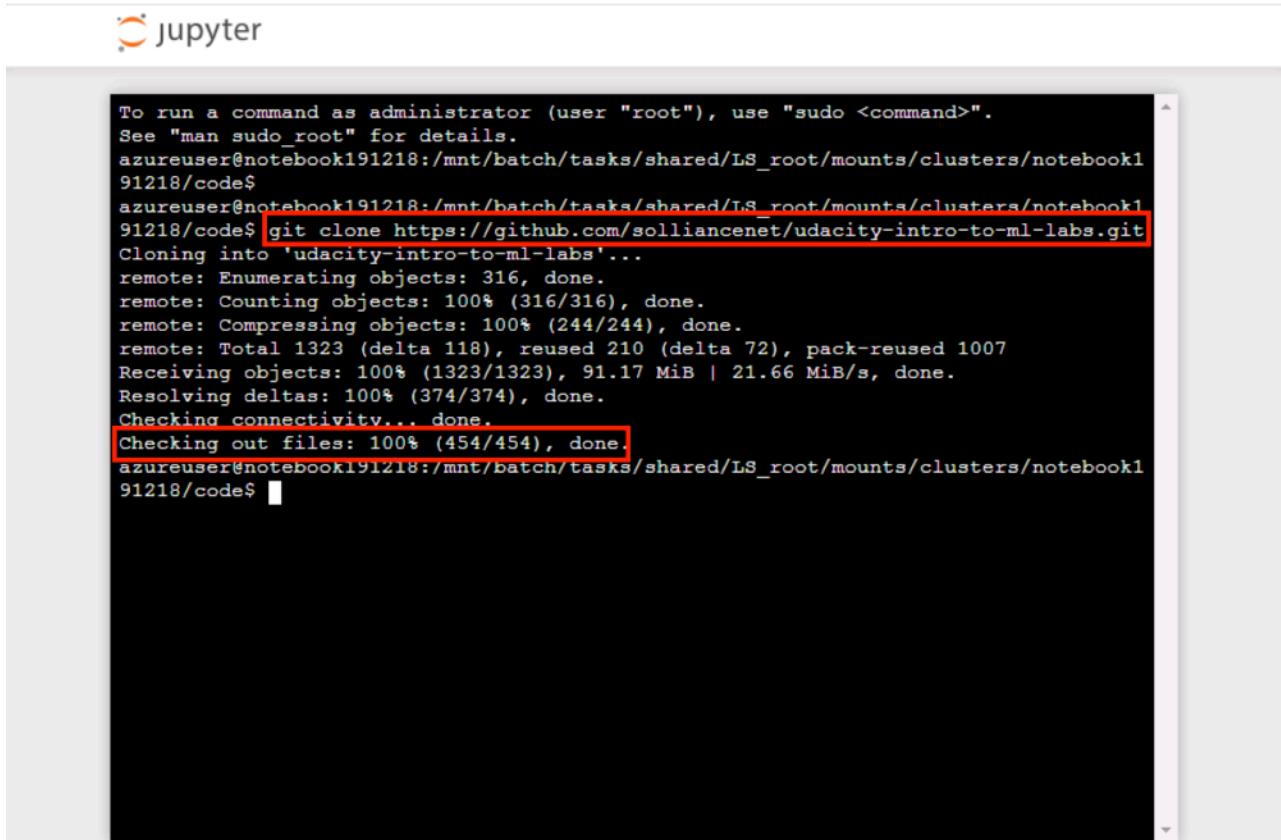
4. From the studio, navigate to **Compute**. Next, for the available Compute Instance, under Application URI select **Jupyter**. Be sure to select **Jupyter** and not **JupyterLab**.

The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a navigation sidebar with various options like New, Home, Author, Notebooks, Automated ML, Designer, Assets, Datasets, Experiments, Pipelines, Models, Endpoints, Manage, Compute, Datastores, and Data Labeling. The 'Compute' option is highlighted with a red box. The main area is titled 'Compute' and shows a list of 'Compute Instances'. The list includes columns for Name, Status, Application URI, Virtual Machine size, and Created on. There are buttons for New, Refresh, Start, Stop, Restart, Delete, Show created by me only, and Search to filter items. A note at the top says 'Compute instance is replacing the Notebook VM. You cannot create new Notebook VMs, but you can still use existing Notebook VMs. Learn More.' One instance is listed: Name is 'quick-starts-ws-191218', Status is 'Running', Application URI is 'JupyterLab Jupyter RStudio SSH' (with 'Jupyter' highlighted by a red box), and Virtual Machine size is 'STANDARD_D3_V2'. Navigation arrows for Prev and Next are also present.

5. From within the Jupyter interface, select **New, Terminal**.

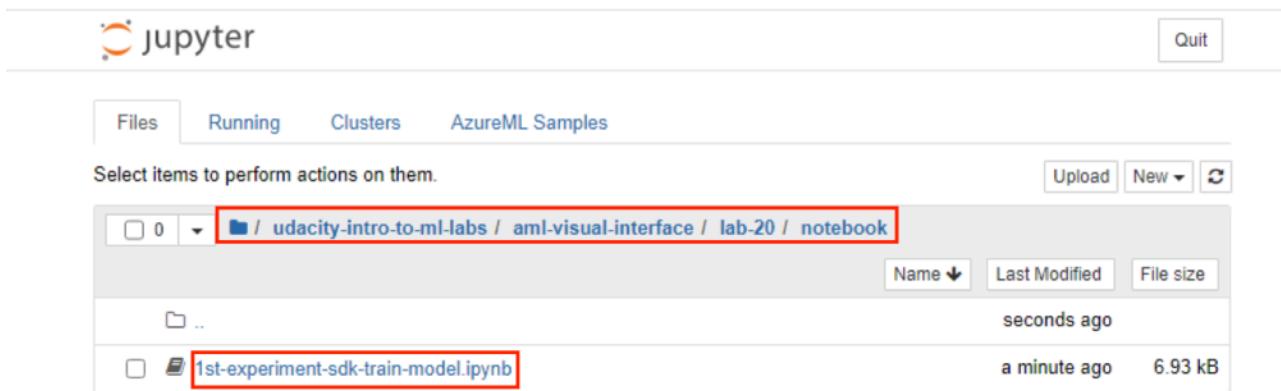
The screenshot shows the Jupyter interface. At the top, there are two browser tabs: one for 'manage.cloudlabs.ai' and another for 'notebook191218.southcentralus.instances.azureml.net'. The main window has a title 'jupyter'. Below the title, there are tabs for Files, Running, Clusters, and AzureML Samples. A sidebar on the left shows a file tree with '0' and 'Users' entries. To the right of the file tree is a 'Name:' input field and a 'New' button. A dropdown menu is open, listing 'Notebook:' options: Python 3, Python 3.6 - AzureML, R, and 'Terminal' (which is highlighted with a red box). Other options in the dropdown include 'Text File' and 'Folder'. A 'Upload' button is also visible in the dropdown.

6. In the new terminal window run the following command and wait for it to finish:
`git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git\`



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$ azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$ git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git
Cloning into 'udacity-intro-to-ml-labs'...
remote: Enumerating objects: 316, done.
remote: Counting objects: 100% (316/316), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 1323 (delta 118), reused 210 (delta 72), pack-reused 1007
Receiving objects: 100% (1323/1323), 91.17 MiB | 21.66 MiB/s, done.
Resolving deltas: 100% (374/374), done.
Checking connectivity... done.
Checking out files: 100% (454/454), done.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$
```

7. From within the Jupyter interface, navigate to directory `udacity-intro-to-ml-labs/aml-visual-interface/lab-20/notebook` and open `1st-experiment-sdk-train-model.ipynb`. This is the Python notebook you will step through executing in this lab.



8. Follow the instructions within the notebook to complete the exercise.

Exercise 2: Open Experiments in the portal

1. Within Azure Machine Learning Studio, select **Experiments** in the left-hand menu, then select the **diabetes-experiment** submitted by the notebook you executed in the previous lab (19).

The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top navigation bar includes 'Preview' (orange), 'Microsoft Azure Machine Learning', and a breadcrumb path 'intro-to-ml-workspace > Experiments'. On the left, a sidebar menu lists various options: 'New', 'Home', 'Author', 'Notebooks', 'Automated ML', 'Designer', 'Assets', 'Datasets', 'Experiments' (which is highlighted with a red box), 'Pipelines', 'Models', 'Endpoints', 'Manage', and 'Compute'. The main content area is titled 'Experiments' and shows a table with two columns: 'Experiment' and 'Latest run'. A single row is visible, containing 'diabetes-experiment' and '10'. There are also 'Add filter' and 'View archived experiments' buttons. Navigation arrows for 'Prev' and 'Next' are at the bottom right.

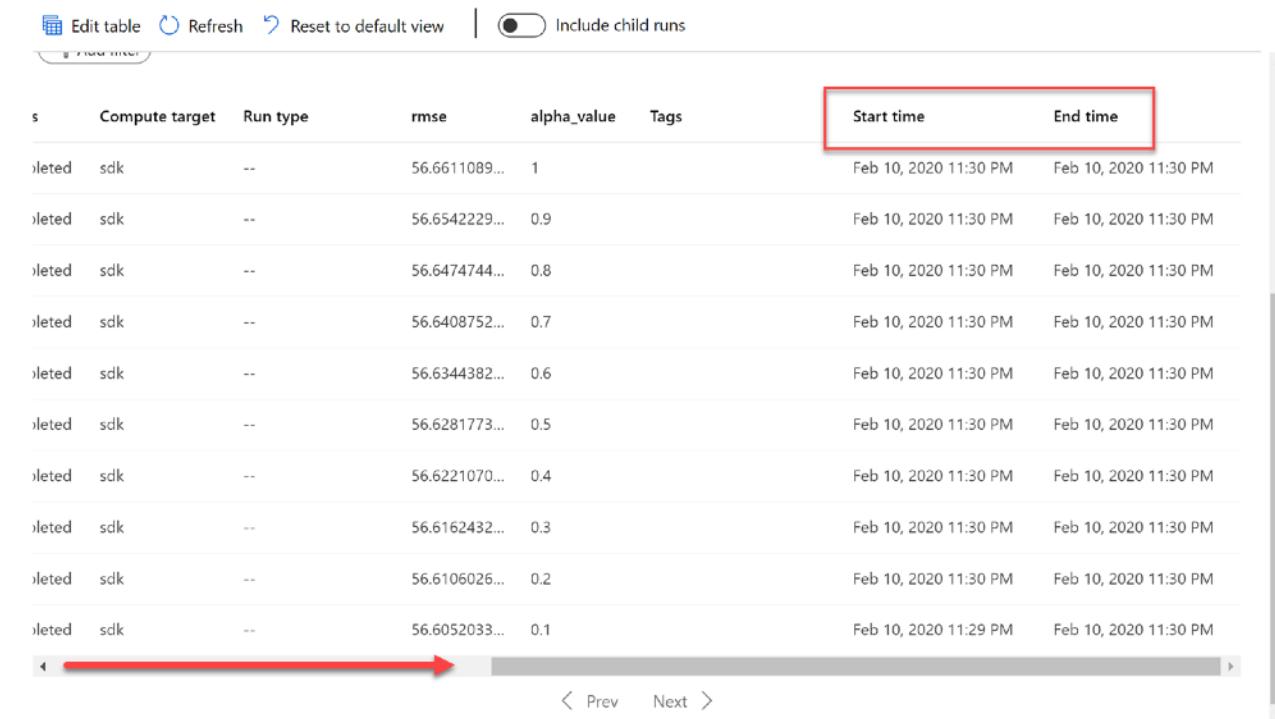
Experiment	Latest run
diabetes-experiment	10

2. Here you can view details about the experiment and each of its runs, which created a new version of the model.

3. Select **Edit table** in the top toolbar. In the Edit table dialog that appears, add the **End time** and **Start time** columns to the Selected columns list, then select **Save**.

Run number	Run ID	Created time	Submitted by
10	5143.....	Feb 10, 2020 11:30 PM	Joel Huler
9	7562.....	Feb 10, 2020 11:30 PM	Joel Huler
8	a718.....	Feb 10, 2020 11:30 PM	Joel Huler
7	aa90.....	Feb 10, 2020 11:30 PM	Joel Huler

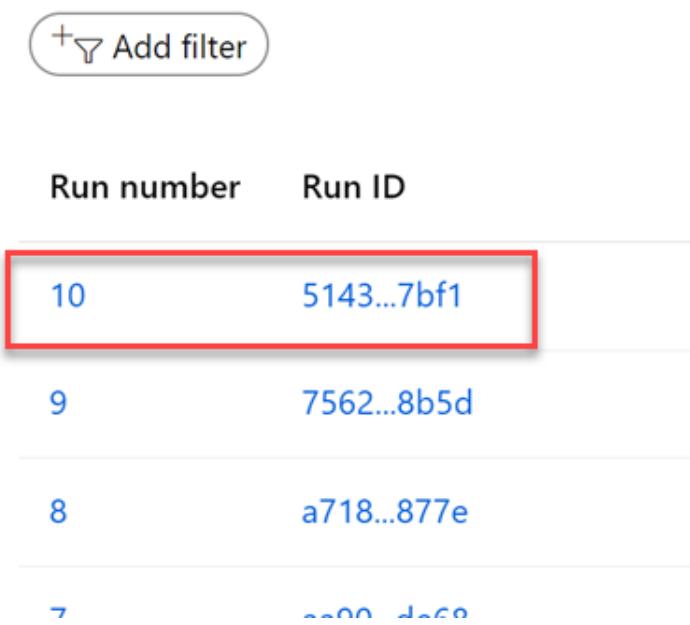
Depending on your screen resolution, you might need to scroll down the table to see the bottom horizontal scrollbar. When you scroll all the way to the right, you will see the new columns you added.



A screenshot of a table interface. At the top, there are buttons for 'Edit table', 'Refresh', 'Reset to default view', and a toggle for 'Include child runs'. The table has columns: 's', 'Compute target', 'Run type', 'rmse', 'alpha_value', 'Tags', 'Start time', and 'End time'. The 'Start time' and 'End time' columns are highlighted with a red border. Below the table is a horizontal scrollbar with a red arrow pointing to the right. At the bottom of the table area are navigation buttons: '< Prev', 'Next >'.

s	Compute target	Run type	rmse	alpha_value	Tags	Start time	End time
selected	sdk	--	56.6611089...	1		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6542229...	0.9		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6474744...	0.8		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6408752...	0.7		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6344382...	0.6		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6281773...	0.5		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6221070...	0.4		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6162432...	0.3		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6106026...	0.2		Feb 10, 2020 11:30 PM	Feb 10, 2020 11:30 PM
selected	sdk	--	56.6052033...	0.1		Feb 10, 2020 11:29 PM	Feb 10, 2020 11:30 PM

4. Select either the **Run number** or the **Run ID** of one of the runs to view its details. Both links on a run display the same dialog.



A screenshot of a table showing run details. The table has two columns: 'Run number' and 'Run ID'. The first row is highlighted with a red box. The 'Run number' column contains values 10, 9, 8, and 7. The 'Run ID' column contains values 5143...7bf1, 7562...8b5d, a718...877e, and -+--+-+--+-. The entire table is enclosed in a rounded rectangle.

Run number	Run ID
10	5143...7bf1
9	7562...8b5d
8	a718...877e
7	-+--+-+--+-

5. The **Details** tab shows you more detailed information about each run, including the run time and metrics.

Run 10 ✓ Completed

↻ Refresh ↻ Resubmit ✖ Cancel

Details Metrics Images Child runs Outputs + logs Snapshot Raw JSON Explanations (Preview)

Properties

Status
Completed

Created
Feb 10, 2020 11:30 PM

Duration
4.270s

Compute target
sdk

Run ID
51436fd3-92f8-40fb-bef2-839b99477bf1

Run number
10

Script name
--

Metrics

rmse	56.66110898499056
alpha_value	1

6. Select the **Outputs + logs** tab. You see the **.pkl** file for the model that was uploaded to the run during each training iteration. This lets you download the model file rather than having to retrain it manually.

Run 10 ✓ Completed

↻ Refresh ↻ Resubmit ✖ Cancel

Details Metrics Images Child runs Outputs + logs Snapshot

🔍

 model_alpha_1.0.pkl

Next Steps

Congratulations! You have just learned how to use the Azure Machine Learning SDK to help you explain what influences the predictions a model makes. You can now return to the Udacity portal to continue with the lesson.

Lab 21 - Deploy a trained model as a webservice

In previous lessons, we spent much time talking about training a machine learning model, which is a multi-step process involving data preparation, feature engineering, training, evaluation, and model selection. The model training process can be very compute-intensive, with training times spanning across many hours, days, or weeks depending on the amount of data, type of algorithm used, and other factors. A trained model, on the other hand, is used to make decisions on new data quickly. In other words, it infers things about new data it is given based on its training. Making these decisions on new data on-demand is called real-time inferencing.

Overview

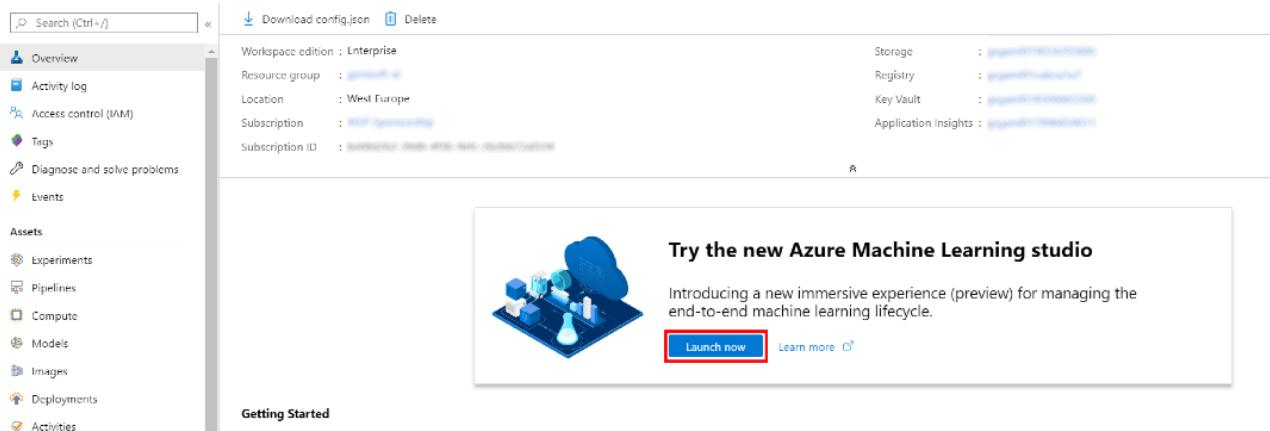
In this lab, you learn how to deploy a trained model that can be used as a webservice, hosted on an Azure Kubernetes Service (AKS) cluster. This process is what enables you to use your model for real-time inferencing.

The Azure Machine Learning designer simplifies the process by enabling you to train and deploy your model without writing any code.

Exercise 1: Open a sample training pipeline

Task 1: Open the pipeline authoring editor

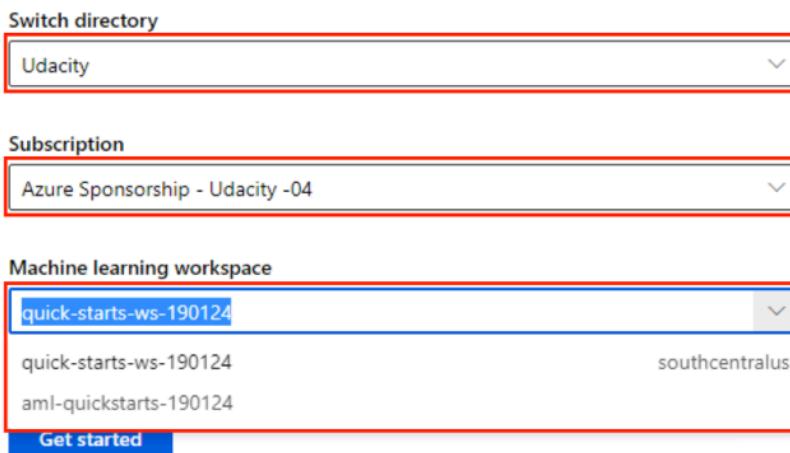
1. In [Azure portal](#), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.



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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



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**.

4. From the studio, select **Designer** in the left-hand menu. Next, select **Sample 1: Regression - Automobile Price Prediction (Basic)** under the **New pipeline** section. This will open a **visual pipeline authoring editor**.

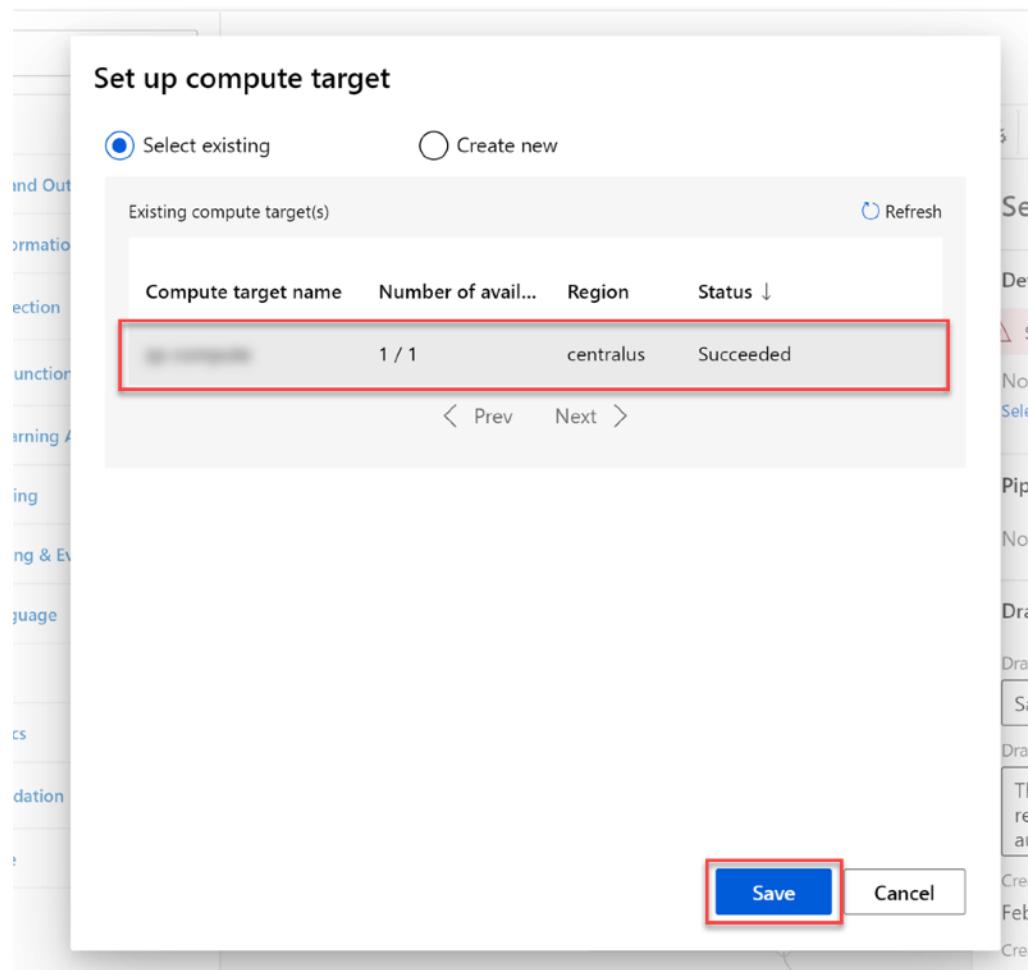
The screenshot shows the Microsoft Azure Machine Learning Studio Designer interface. The left sidebar has a 'Designer' icon highlighted with a red box. The main area shows 'New pipeline' options: 'Easy-to-use prebuilt modules' (with 'Sample 1: Regression - Automobile Price Prediction...' highlighted with a red box) and 'Pipelines' (with 'Sample 2: Re...' partially visible).

Task 2: Setup the compute target

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

The screenshot shows the Microsoft Azure Machine Learning Studio Designer interface with a pipeline diagram. A tooltip for 'Select Compute Target' is open, showing the message 'No compute target selected' and a link to 'Select compute target'.

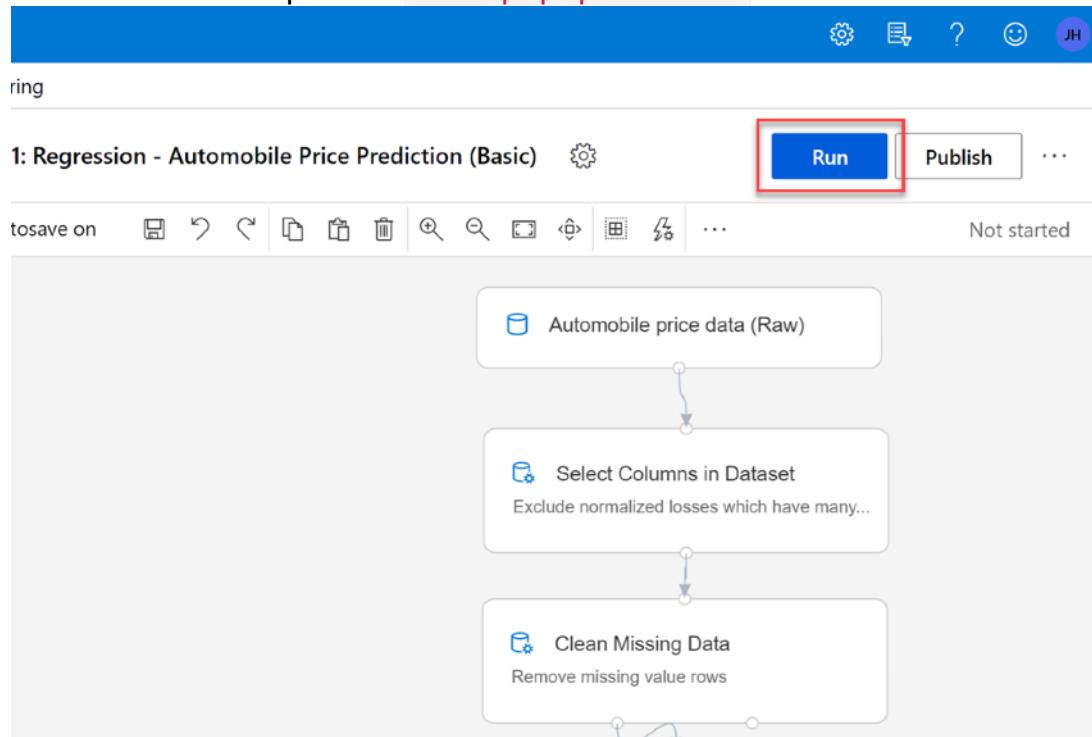
2. In the **Set up compute target** editor, select the existing compute target, then select **Save**.



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.

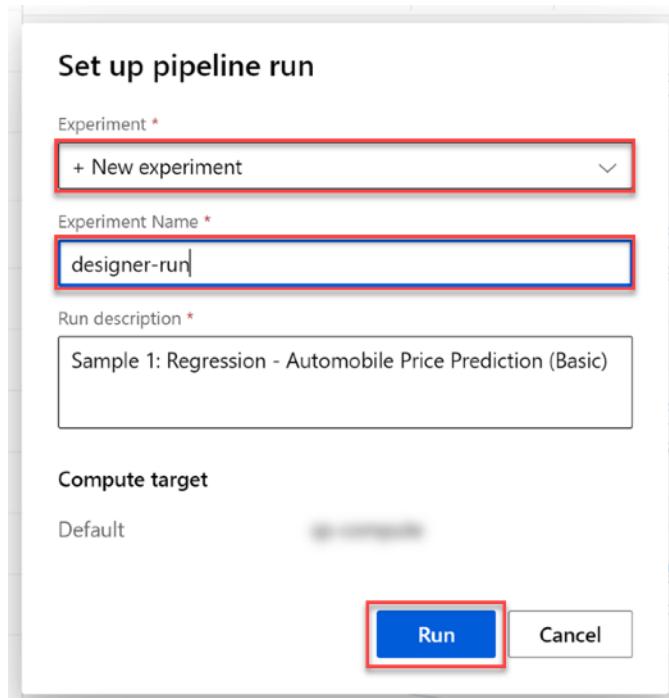
Task 3: Create a new experiment and submit the pipeline

1. Select **Submit** to open the **Set up pipeline run editor**.



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

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

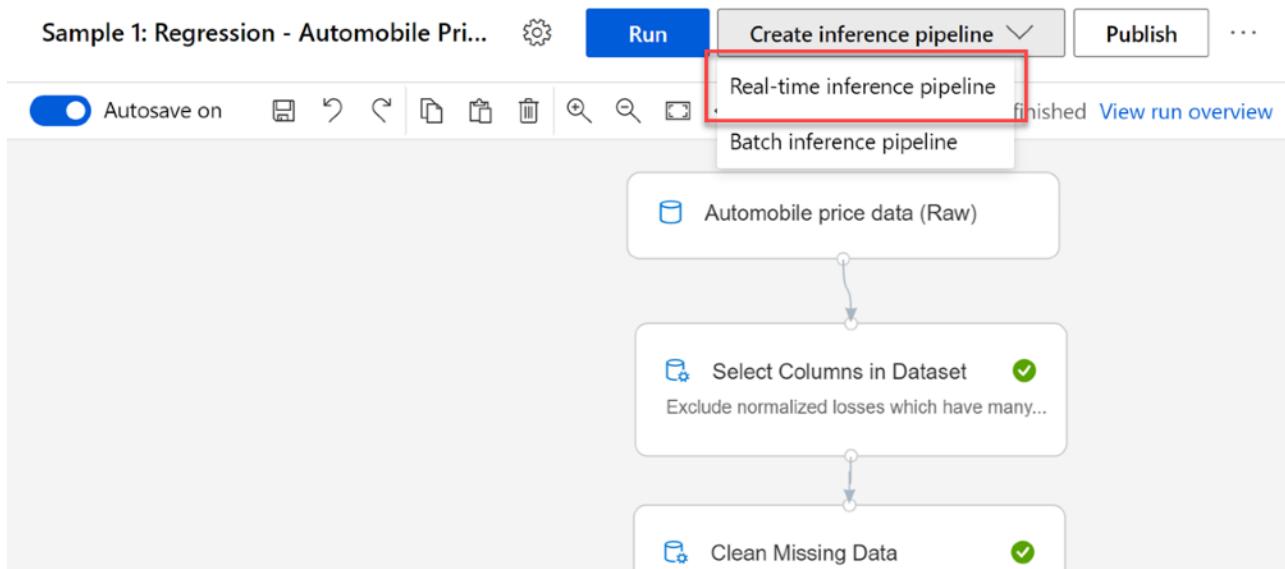


3. Wait for the pipeline run to complete. It will take around **10 minutes** to complete the run.

Exercise 2: Real-time inference pipeline

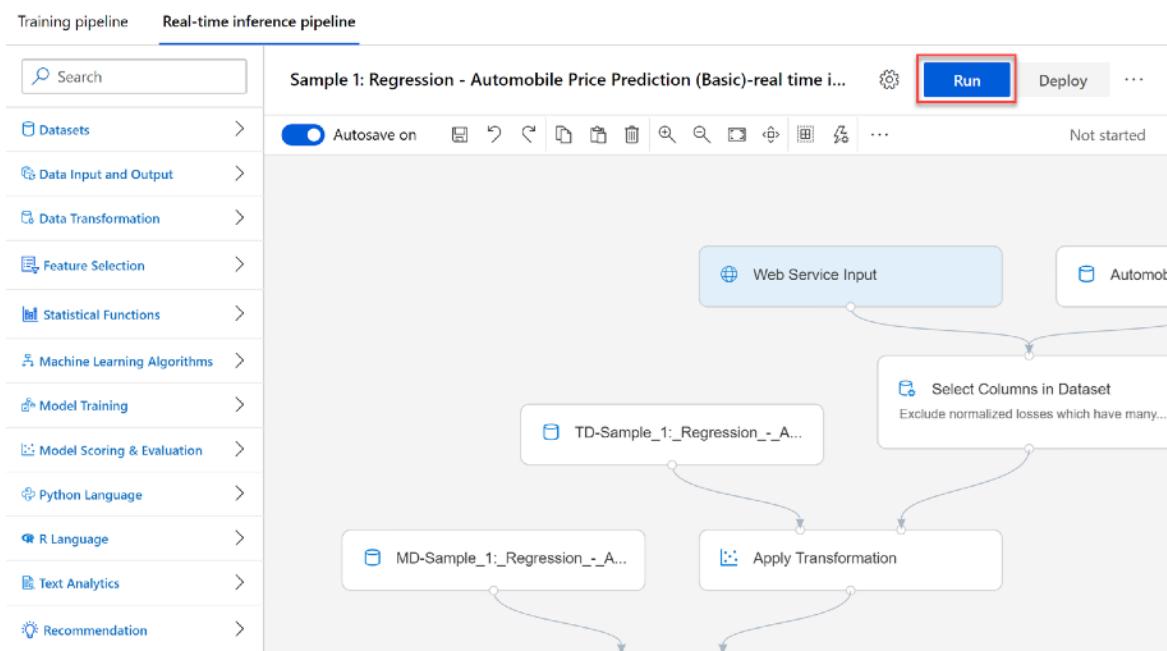
Task 1: Create pipeline

1. Select **Create inference pipeline**, then select **Real-time inference pipeline** from the list to create a new inference pipeline.



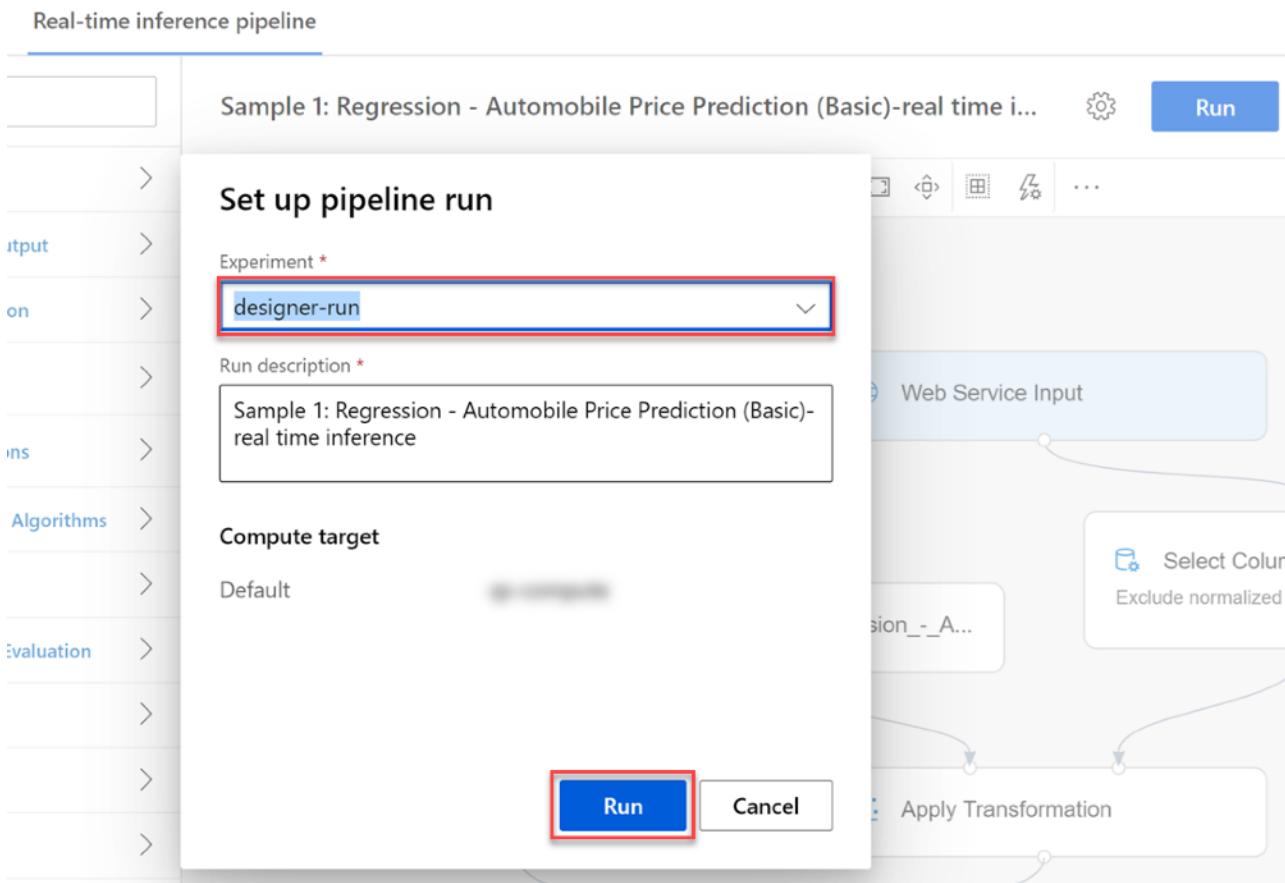
Task 2: Submit the pipeline

1. Select **Submit** to open the **Set up pipeline run** editor.



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

2. In the **Setup pipeline run** editor, select **Select existing**, then select the experiment you created in an earlier step: **designer-run**. Select **Submit** to start the pipeline.

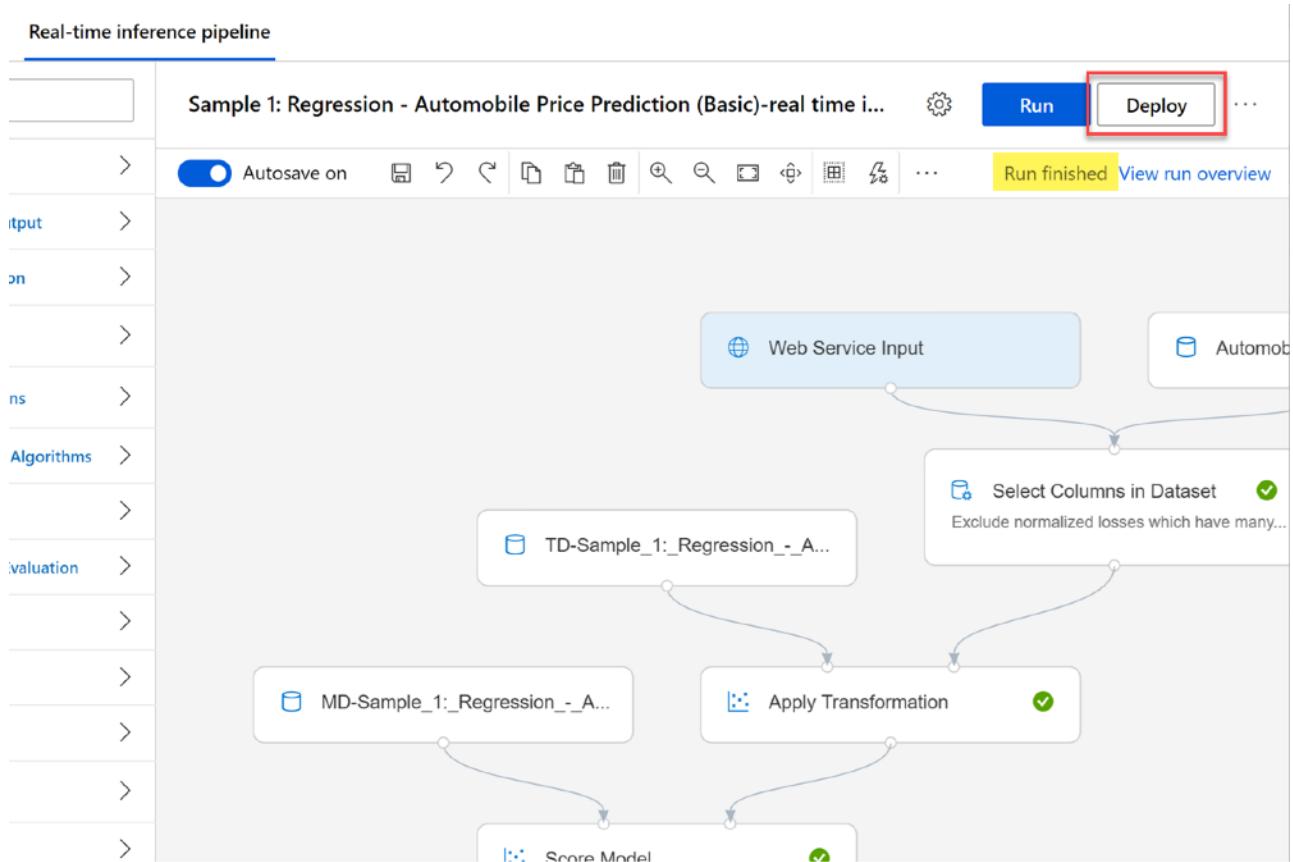


3. Wait for pipeline run to complete. It will take around **7 minutes** to complete the run.

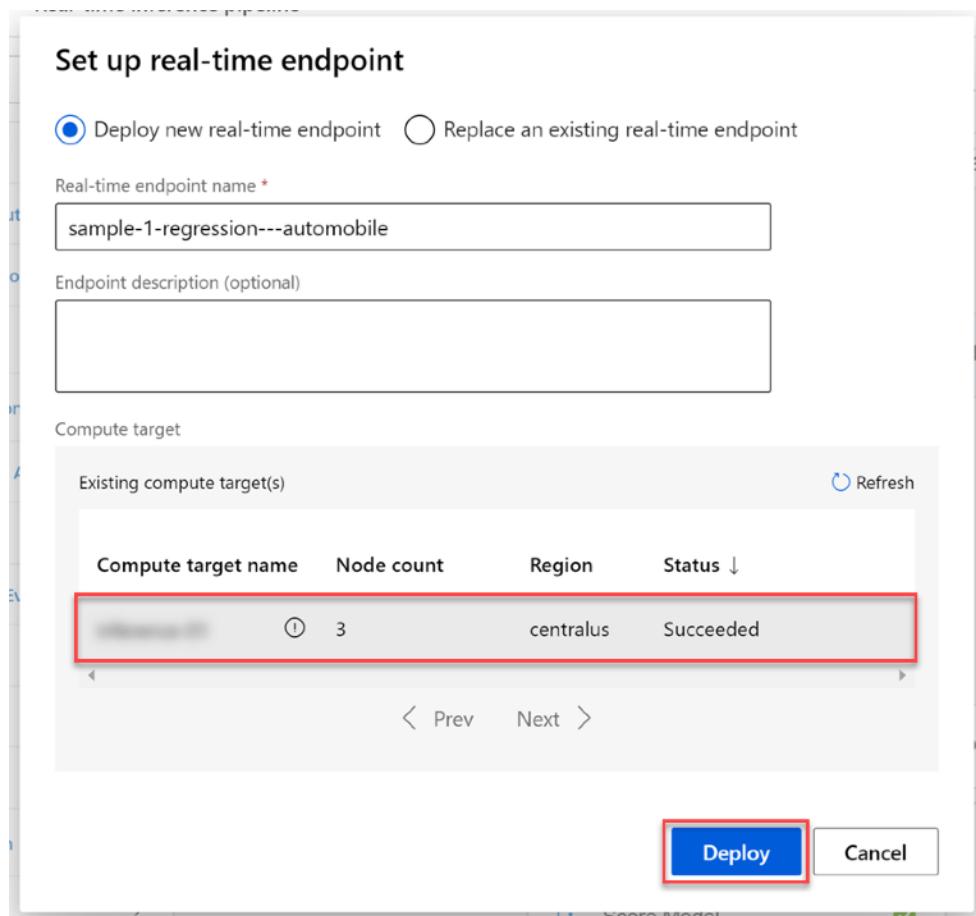
Exercise 3: Deploy web service on Azure Kubernetes Service compute

Task 1: Deploy the web service

- After the inference pipeline run is finished, select **Deploy** to open the **Set up real-time endpoint** editor.



2. In the **Set up real-time endpoint** editor, select your **existing compute target**, then select **Deploy**.



3. Wait for the deployment to complete. The status of the deployment can be observed above the **Pipeline Authoring Editor**.

The screenshot shows the Pipeline Authoring Editor interface. At the top, a green bar displays a success message: 'Deploy: Succeeded [view real-time endpoint](#)'. Below this, there are two tabs: 'Training pipeline' and 'Real-time inference pipeline', with 'Real-time inference pipeline' selected. On the left, there's a search bar labeled 'Search' and two sections: 'Datasets' and 'Data Input and Output'. On the right, the 'Real-time inference pipeline' tab is open, showing 'Sample 1: Regression - Autom' and a toolbar with 'Autosave on' (switched on), a save icon, and other controls.

Task 2: Review deployed web service

1. To view the deployed web service, select the **Endpoints** section in your Azure Portal Workspace.
2. Select the deployed web service: **sample-1-regression---automobile** to open the deployment details page.

The screenshot shows the Microsoft Azure Machine Learning workspace interface. At the top, there's a blue header bar with the text "Microsoft Azure Machine Learning". Below it is a navigation sidebar with several options: "Preview" (highlighted in orange), "New", "Home", "Author" (with "Notebooks", "Automated ML", and "Designer" listed), "Assets" (with "Datasets", "Experiments", "Pipelines", "Models", and "Endpoints" highlighted with a red box), and "Manage". The main content area has a breadcrumb path "intro-to-ml-workspace > Endpoints". It displays a table titled "Endpoints" with two rows. The first row, "sample-1-regression---automobile", is highlighted with a red box around its "Name" column. The second row is "nyc-taxi-srv" with the description "NYC Taxi Fare Pre...". There are "Refresh" and "Delete" buttons below the table. The "Real-time endpoints" tab is active, while the "Pipeline endpoints" tab is inactive.

Name	Description	Created On
sample-1-regression---automobile		[Redacted]
nyc-taxi-srv	NYC Taxi Fare Pre...	[Redacted]

Note: you have to select the text of the service name to open the deployment details page

Task 3: Review how to consume the deployed web service

1. Select the **Consume** tab to observe the following information:
 1. **Basic consumption info** displays the **REST endpoint**, **Primary key**, and **Secondary key**.
 2. **Consumption option** shows code samples in **C#**, **Python**, and **R** on how to call the endpoint to consume the webservice.

sample-1-regression---automobile

The screenshot shows the 'Consume' tab selected in the top navigation bar. Below it, the 'Basic consumption info' section is displayed, containing the REST endpoint URL and options for using a key or token. The 'Consumption option' section is also shown, featuring tabs for C#, Python, and R, with the Python tab currently active, displaying sample code for making a request to the service.

```
1 import urllib.request
2 import json
3 import os
4 import ssl
5
6 def allowSelfSignedHttps(allowed):
7     # bypass the server certificate verification on client side
8     if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and getattr(ssl, '_create_unverified_context', None):
9         ssl._create_default_https_context = ssl._create_unverified_context
10
11 allowSelfSignedHttps(True) # this line is needed if you use self-signed certificate in your scoring service.
12
13 data = {
14     "Inputs": {
15         "WebServiceInput0": [
16             [

```

Next Steps

Congratulations! You have just learned how to train and deploy a model to an Azure Kubernetes Service (AKS) cluster for real-time inferencing. You can now return to the Udacity portal to continue with the lesson.

Lab 22 - Training and deploying a model from a notebook running in a Compute Instance

So far, the Managed Services for Azure Machine Learning lesson has covered **compute instance** and the benefits it provides through its fully managed environment containing everything you need to run Azure Machine Learning.

The compute instance provides a comprehensive set of capabilities that you can use directly within a python notebook or python code including:

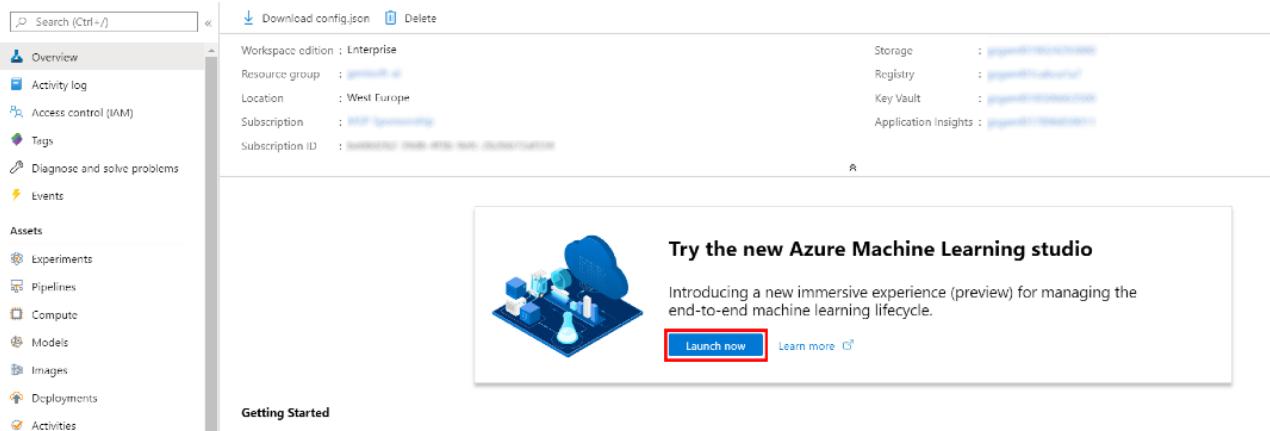
- Creating a **Workspace** that acts as the root object to organize all artifacts and resources used by Azure Machine Learning.
- Creating **Experiments** in your Workspace that capture versions of the trained model along with any desired model performance telemetry. Each time you train a model and evaluate its results, you can capture that run (model and telemetry) within an Experiment.
- Creating **Compute** resources that can be used to scale out model training, so that while your notebook may be running in a lightweight container in Azure Notebooks, your model training can actually occur on a powerful cluster that can provide large amounts of memory, CPU or GPU.
- Using **Automated Machine Learning (AutoML)** to automatically train multiple versions of a model using a mix of different ways to prepare the data and different algorithms and hyperparameters (algorithm settings) in search of the model that performs best according to a performance metric that you specify.
- Packaging a Docker **Image** that contains everything your trained model needs for scoring (prediction) in order to run as a web service.
- Deploying your Image to either Azure Kubernetes or Azure Container Instances, effectively hosting the **Web Service**.

Overview

In this lab, you start with a model that was trained using Automated Machine Learning. Learn how to use the Azure ML Python SDK to register, package, and deploy the trained model to Azure Container Instances (ACI) as a scoring web service. Finally, test the deployed model (1) by make direct calls on service object, (2) by calling the service end point (Scoring URI) over http.

Exercise 1: Run the Notebook for this Lab

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

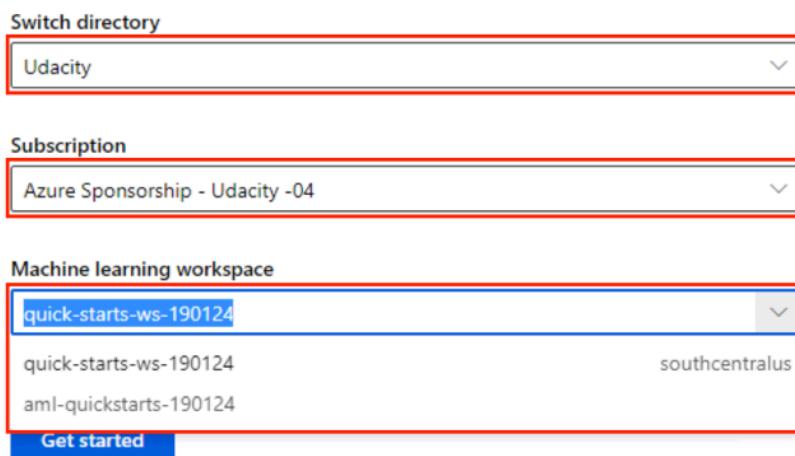


The screenshot shows the Azure Machine Learning workspace overview page. On the right, there's a promotional section for the new Azure Machine Learning studio, featuring a 3D icon of a lab setup and the text "Try the new Azure Machine Learning studio". Below it is a sub-section titled "Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle." with a "Launch now" button (which is highlighted with a red box) and a "Learn more" link. The left sidebar contains a navigation menu with items such as Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets, Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The "Getting Started" section is also visible.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).



The screenshot shows the "Welcome to the studio!" configuration screen. It has three main dropdown menus: "Switch directory" (set to "Udacity"), "Subscription" (set to "Azure Sponsorship - Udacity -04"), and "Machine learning workspace" (listing "quick-starts-ws-190124", "quick-starts-ws-190124", and "aml-quickstarts-190124"). At the bottom is a blue "Get started" button.

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**.

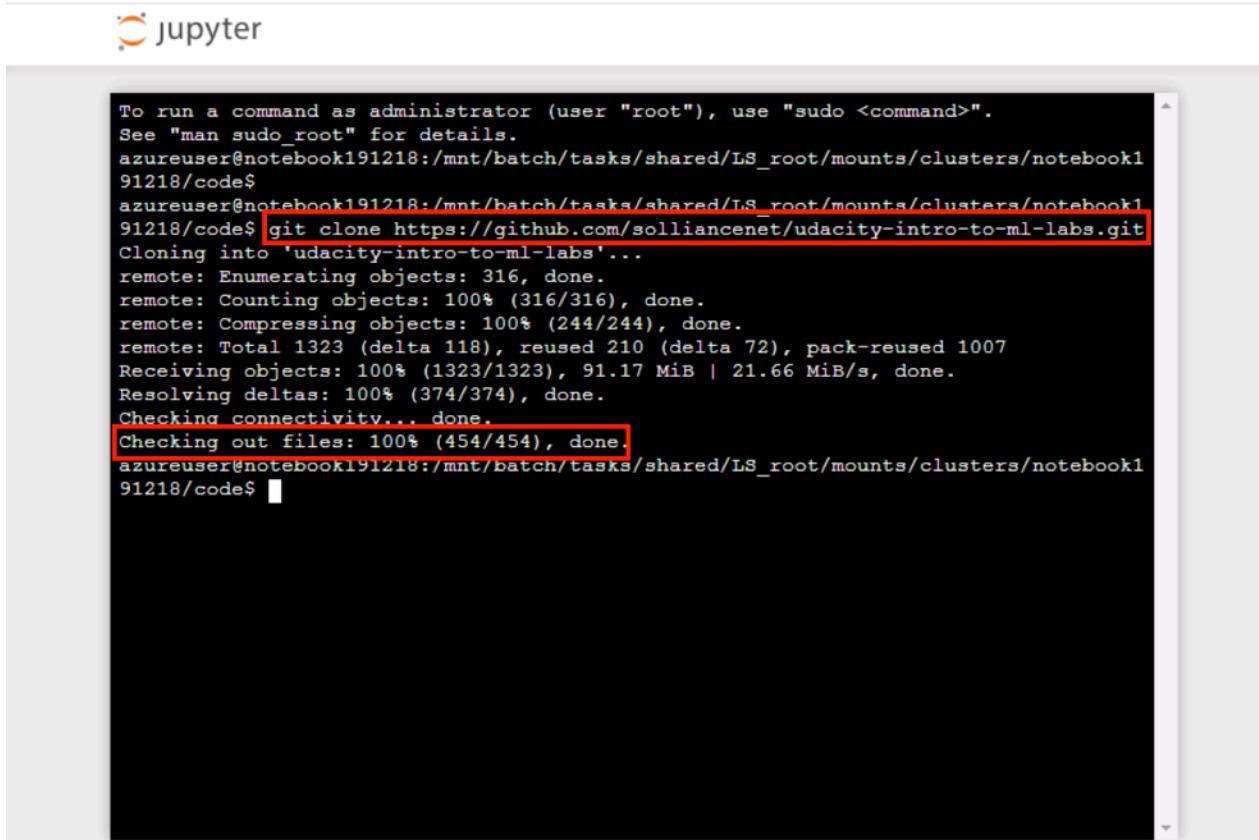
4. From the studio, navigate to **Compute**. Next, for the available Compute Instance, under Application URI select **Jupyter**. Be sure to select **Jupyter** and not **JupyterLab**.

The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a navigation sidebar with various options like New, Home, Author, Notebooks, Automated ML, Designer, Assets, Datasets, Experiments, Pipelines, Models, Endpoints, Manage, Compute (which is highlighted with a red box), Datastores, and Data Labeling. The main area is titled 'Compute' and shows a list of 'Compute Instances'. The list includes columns for Name, Status, Application URI, Virtual Machine size, and Created on. There are buttons for New, Refresh, Start, Stop, Restart, Delete, Show created by me only, and Search to filter items. A note at the top says 'Compute instance is replacing the Notebook VM. You cannot create new Notebook VMs, but you can still use existing Notebook VMs. Learn More.' One instance is listed: Name is 'quick-starts-ws-191218', Status is 'Running', Application URI is 'JupyterLab Jupyter RStudio SSH' (with 'Jupyter' highlighted with a red box), and Virtual Machine size is 'STANDARD_D3_V2'. Navigation arrows for Prev and Next are also present.

5. From within the Jupyter interface, select **New, Terminal**.

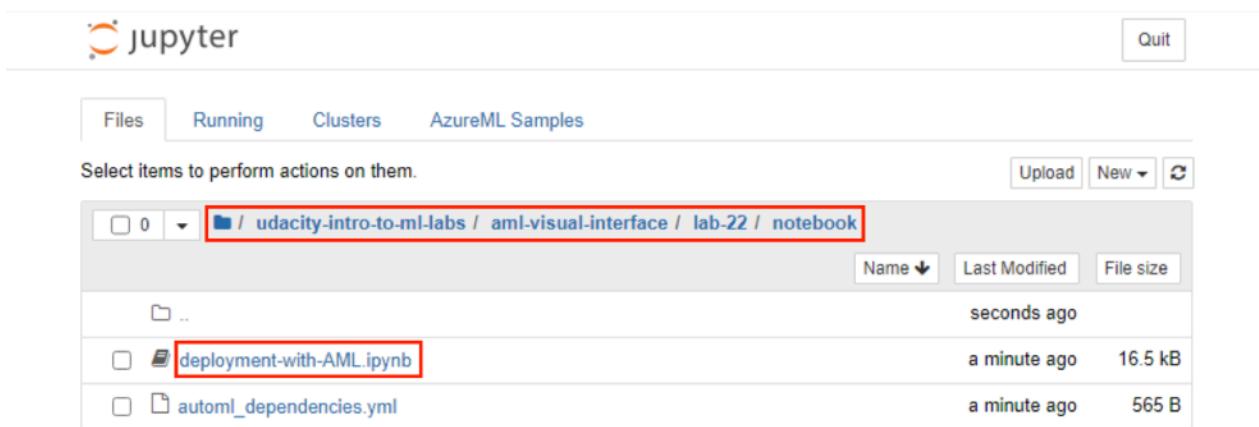
The screenshot shows the Jupyter interface. At the top, there are two browser tabs: one for 'manage.cloudlabs.ai' and another for 'notebook191218.southcentralus.instances.azureml.net'. The main window has a title 'jupyter'. Below the title, there are tabs for Files, Running, Clusters, and AzureML Samples. A sidebar on the left shows a file tree with '0' and 'Users' entries. A context menu is open over the '0' entry, listing options: Upload, New (with a dropdown menu showing Notebook: Python 3, Python 3.6 - AzureML, R, and Other: Text File, Folder, Terminal). The 'Terminal' option is highlighted with a red box and has a mouse cursor pointing at it.

6. In the new terminal window run the following command and wait for it to finish:
`git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git`



To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code\$ git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git
Cloning into 'udacity-intro-to-ml-labs'...
remote: Enumerating objects: 316, done.
remote: Counting objects: 100% (316/316), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 1323 (delta 118), reused 210 (delta 72), pack-reused 1007
Receiving objects: 100% (1323/1323), 91.17 MiB | 21.66 MiB/s, done.
Resolving deltas: 100% (374/374), done.
Checking connectivity... done.
Checking out files: 100% (454/454), done.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code\$

7. From within the Jupyter interface, navigate to directory `udacity-intro-to-ml-labs/aml-visual-interface/lab-22/notebook` and open `deployment-with-AML.ipynb`. This is the Python notebook you will step through executing in this lab.



8. In the Setup portion of the notebook, you will be asked to provide values for `subscription_id`, `resource_group`, `workspace_name`, and `workspace_region`. To find these, open your Azure Machine Learning workspace in the Azure portal and copy the values as shown:

The screenshot shows the Azure Machine Learning workspace setup page. It includes instructions for providing subscription information and navigating the Azure portal. Below the instructions is a code cell containing Python code for setting environment variables. The workspace details pane on the right is highlighted with four red boxes, pointing to the workspace name, resource group, location, and subscription ID.

```
In [ ]: #Provide the Subscription ID of your existing Azure subscription  
subscription_id = "" # <- needs to be the subscription within the Azure resource group for this lesson  
  
#Provide values for the existing Resource Group  
resource_group = "" # <- enter the name of your Azure Resource Group  
  
#Provide the Workspace Name and Azure Region of the Azure Machine Learning Workspace  
workspace_name = "" # <- enter the name of the Azure Machine Learning workspace  
workspace_region = "eastus" # <- region of your Azure Machine Learning workspace
```

9. Follow the instructions within the notebook to complete the lab.

Next Steps

Congratulations! You have just learned how to use the Jupyter application on a compute instance to deploy a trained model to Azure Container Instances (ACI) for real-time inferencing. You can now return to the Udacity portal to continue with the lesson.

Lab 23 - Explaining models: Model interpretability with Azure Machine Learning service

Machine learning interpretability is important in two phases of machine learning development cycle:

- During training: Model designers and evaluators require interpretability tools to explain the output of a model to stakeholders to build trust. They also need insights into the model so that they can debug the model and make decisions on whether the behavior matches their objectives. Finally, they need to ensure that the model is not biased.
- During inferencing: Predictions need to be explainable to the people who use your model. For example, why did the model deny a mortgage loan, or predict that an investment portfolio carries a higher risk?

The [Azure Machine Learning Interpretability Python SDK](#) incorporates technologies developed by Microsoft and proven third-party libraries (for example, SHAP and LIME). The SDK creates a common API across the integrated libraries and integrates Azure Machine Learning services. Using this SDK, you can explain machine learning models globally on all data, or locally on a specific data point using the state-of-art technologies in an easy-to-use and scalable fashion.

Overview

In this lab, we will be using a subset of NYC Taxi & Limousine Commission - green taxi trip records available from [Azure Open Datasets](#). The data is enriched with holiday and weather data. We will use data transformations and the GradientBoostingRegressor algorithm from the scikit-learn library to train a regression model to predict taxi fares in New York City based on input features such as, number of passengers, trip distance, datetime, holiday information and weather information.

The primary goal of this quickstart is to explain the predictions made by our trained model with the various [Azure Model Interpretability](#) packages of the Azure Machine Learning Python SDK.

Exercise 1: Run the Notebook for this Lab

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

The screenshot shows the Azure Machine Learning workspace overview page. On the left, there's a sidebar with various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Assets, Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The main area displays workspace details: Edition (Enterprise), Resource group (ml-quickstarts), Location (West Europe), Subscription (ml-quickstarts), and Subscription ID (ml-quickstarts). Below this, there's a 'Getting Started' section with a card titled 'Try the new Azure Machine Learning studio'. The card contains a small 3D model of a city on a circuit board, the text 'Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle.', and two buttons: 'Launch now' (which is highlighted with a red box) and 'Learn more'.

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

Welcome to the studio!

Select a subscription and a workspace to get started or go to the [Azure Portal](#) to create your subscription and workspace. You can switch subscriptions and workspaces at any time. [Learn more](#).

The screenshot shows the 'Welcome to the studio!' configuration screen. It has three main dropdown sections: 'Switch directory' (set to 'Udacity'), 'Subscription' (set to 'Azure Sponsorship - Udacity -04'), and 'Machine learning workspace' (listing 'quick-starts-ws-190124', 'quick-starts-ws-190124', and 'aml-quickstarts-190124'). At the bottom is a blue 'Get started' button.

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**.

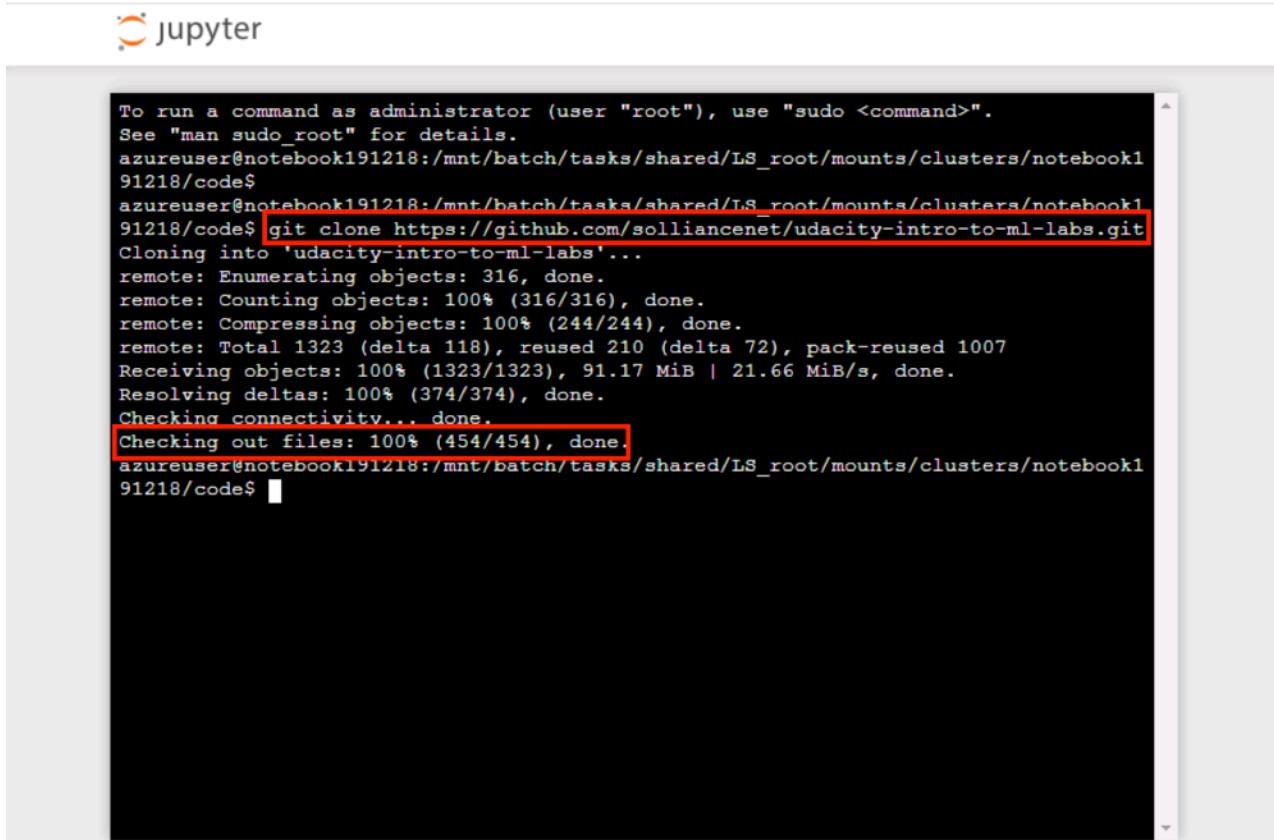
4. From the studio, navigate to **Compute**. Next, for the available Compute Instance, under Application URI select **Jupyter**. Be sure to select **Jupyter** and not **JupyterLab**.

The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a navigation sidebar with various options like New, Home, Author, Notebooks, Automated ML, Designer, Assets, Datasets, Experiments, Pipelines, Models, Endpoints, Manage, Compute (which is highlighted with a red box), Datastores, and Data Labeling. The main area is titled "Compute" and shows a list of "Compute Instances". The list includes columns for Name, Status, Application URI, Virtual Machine size, and Created on. There are buttons for New, Refresh, Start, Stop, Restart, Delete, Show created by me only, and Search to filter items. A note at the top says "Compute instance is replacing the Notebook VM. You cannot create new Notebook VMs, but you can still use existing Notebook VMs. Learn More." One instance is listed: Name is "quick-starts-ws-191218", Status is "Running", Application URI is "JupyterLab Jupyter RStudio SSH" (with "Jupyter" highlighted with a red box), and Virtual Machine size is "STANDARD_D3_V2".

5. From within the Jupyter interface, select **New, Terminal**.

The screenshot shows the Jupyter interface. At the top, there are two browser tabs: one for "manage.cloudlabs.ai" and another for "notebook191218.southcentralus.instances.azureml.net". The main window has a title "jupyter". Below the title, there are tabs for Files, Running, Clusters, and AzureML Samples. A sidebar on the left shows a file structure with "0" and "Users" folders. A context menu is open over the "0" folder, listing options: Upload, New (with a dropdown menu showing Notebook: Python 3, Python 3.6 - AzureML, R, and Other: Text File, Folder, Terminal). The "Terminal" option is highlighted with a red box and has a cursor arrow pointing to it.

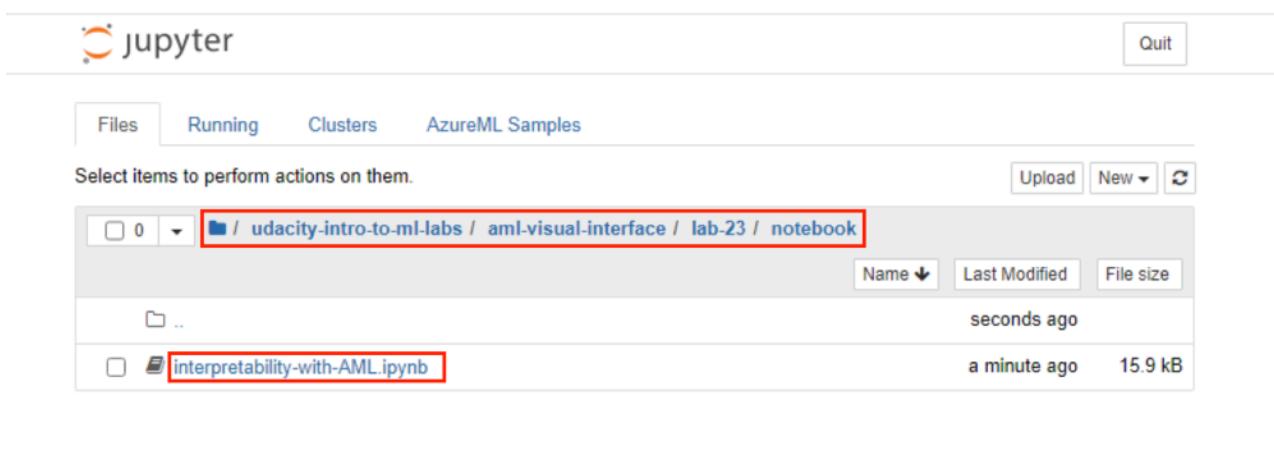
6. In the new terminal window run the following command and wait for it to finish:
`git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git`



The screenshot shows a terminal window titled "jupyter". The command `git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git` is highlighted with a red box. The output of the command is displayed below, showing the progress of cloning the repository from GitHub.

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$ git clone https://github.com/solliancenet/udacity-intro-to-ml-labs.git
Cloning into 'udacity-intro-to-ml-labs'...
remote: Enumerating objects: 316, done.
remote: Counting objects: 100% (316/316), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 1323 (delta 118), reused 210 (delta 72), pack-reused 1007
Receiving objects: 100% (1323/1323), 91.17 MiB | 21.66 MiB/s, done.
Resolving deltas: 100% (374/374), done.
Checking connectivity... done.
Checking out files: 100% (454/454), done.
azureuser@notebook191218:/mnt/batch/tasks/shared/LS_root/mounts/clusters/notebook1
91218/code$
```

7. From within the Jupyter interface, navigate to directory `udacity-intro-to-ml-labs/aml-visual-interface/lab-23/notebook` and open `interpretability-with-AML.ipynb`. This is the Python notebook you will step through executing in this lab.



8. Follow the instructions within the notebook to complete the lab.

Next Steps

Congratulations! You have just learned how to use the Azure Machine Learning SDK to help you explain what influences the predictions a model makes. You can now return to the Udacity portal to continue with the lesson.