CIS5560 Term Project Tutorial

Authors: Nikita Marathe; Nikeeta Akbari; Rohit Tiwari; Zeeshan Khan

Instructor: Jongwook Woo

Date: 05/18/2018

Lab Tutorial

Nikita Marathe (nmarath@calstatela.edu)

Nikeeta Akbari (nakbari2@calstatela.edu)

Rohit Tiwari (rtiwari2@calstatela.edu)

Zeeshan Khan (zkhan7@calstatela.edu)

05/18/2018

Home Mortgage Prediction Model using Azure ML

Objectives

The main objective of this tutorial is to build a machine learning predictive model to predict if the home mortgage application will be approved or denied. In this hands-on lab, you will learn how to:

- Create a new experiment and import data
- Pre-process data
- Perform regression and classification using various algorithms

Pre-requisite

- An Azure ML account
- A web browser and internet connection
- Dataset for Home Mortgage Disclosure Act. Available to download here:

  https://www.consumerfinance.gov/data-research/hmda/explore

- SQL source code used for data preprocessing. Available to download here:

Platform Spec

- Maximum number of modules per experiment: 100
- Maximum storage space: 10 GB
- Execution/ Performance: Single Node

Create an Azure ML Account

Azure ML offers a free-tier account, which you can use to complete this tutorial.

Sign Up for a Microsoft Account

- If you do not already have a Microsoft account, sign up for one at https://signup.live.com/ . You don't need to use your school email account to sign up but you can use any email account.

Sign Up for a Free Azure ML Account

1. Browse to http://bit.ly/azureml_login and click **Get started now**.
2. When prompted, choose the option to sign in, and sign in with your Microsoft account credentials.
3. On the **Welcome** page, watch the overview video if you want to see an introduction to Azure ML Studio. Then close the **Welcome** page by clicking the checkmark icon.

Creating an Azure ML Experiment

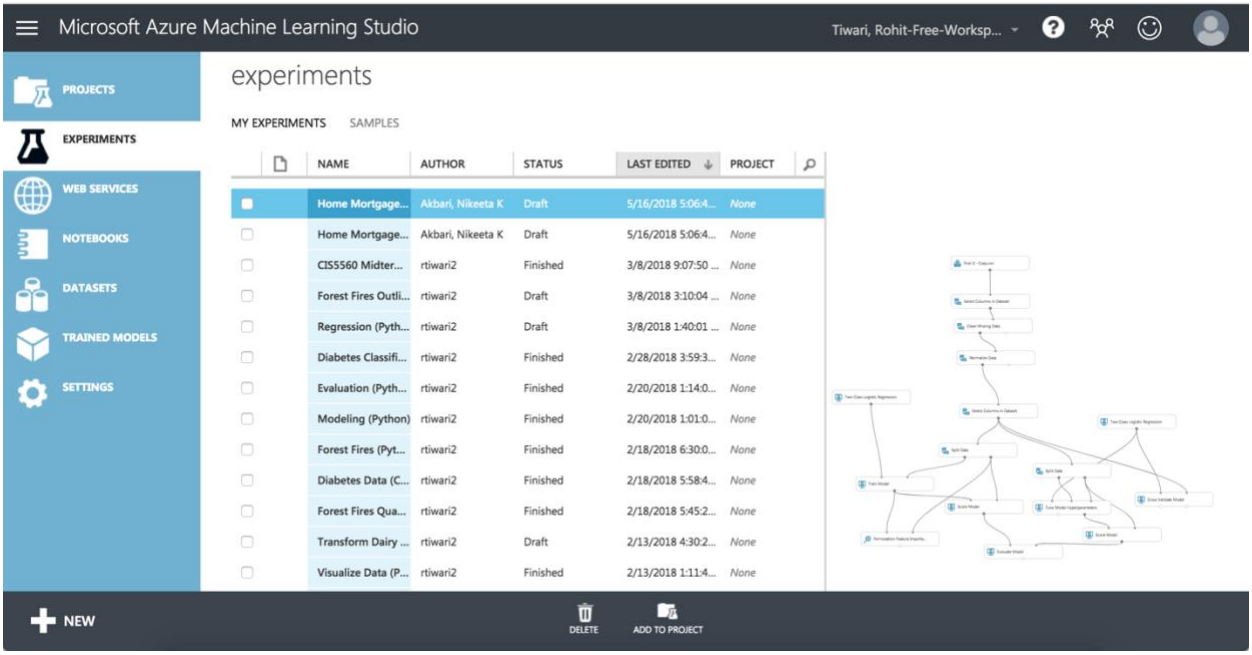
Azure ML enables you to create experiments in which you can manipulate data, create predictive models, and visualize the results. In this tutorial, you will create a simple experiment in which you will explore a sample dataset that contains details on the Home Mortgage Application, from which you can predict whether a new applicant is fit to apply for home mortgage. **We will be using Two-Class Logistic Regression Model and Two-Class Decision Forest Model.**

Sign into Azure ML Studio

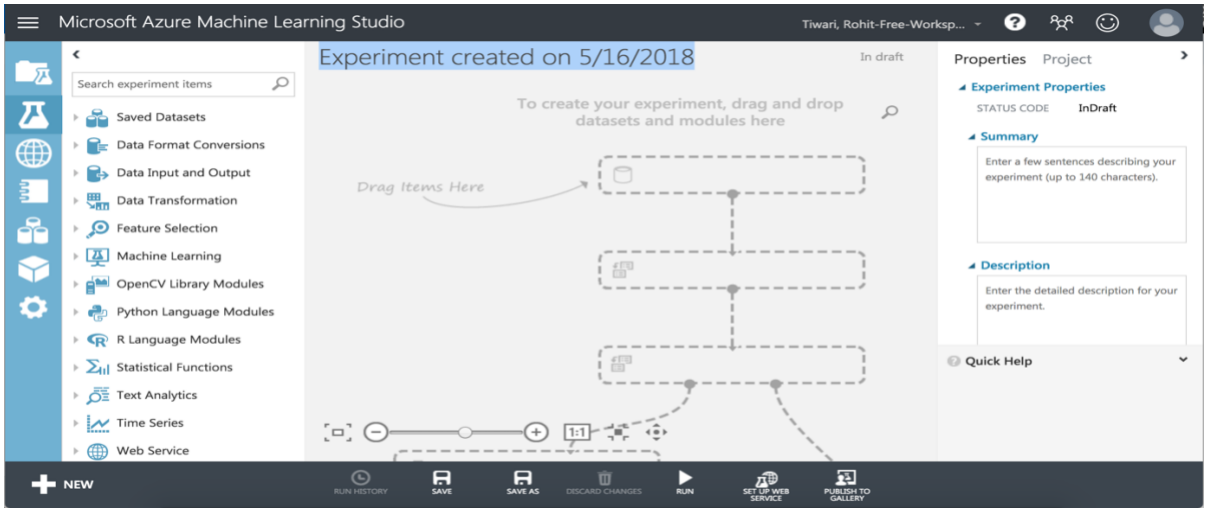
1. Open a browser and browse to https://studio.azureml.net.
2. Click **Sign In** and sign in using the Microsoft account associated with your free Azure ML account.

3. If the Welcome page is displayed, close it by clicking the **OK** icon (which looks like a checkmark). Then, if the New page (containing a collection of Microsoft samples) is displayed, close it by clicking the Close icon (which looks like an X).

4. You should now be in Azure ML Studio with the Experiments page selected, which looks like the following image (if not, click the menu button at the top left corner of the page and select studio).



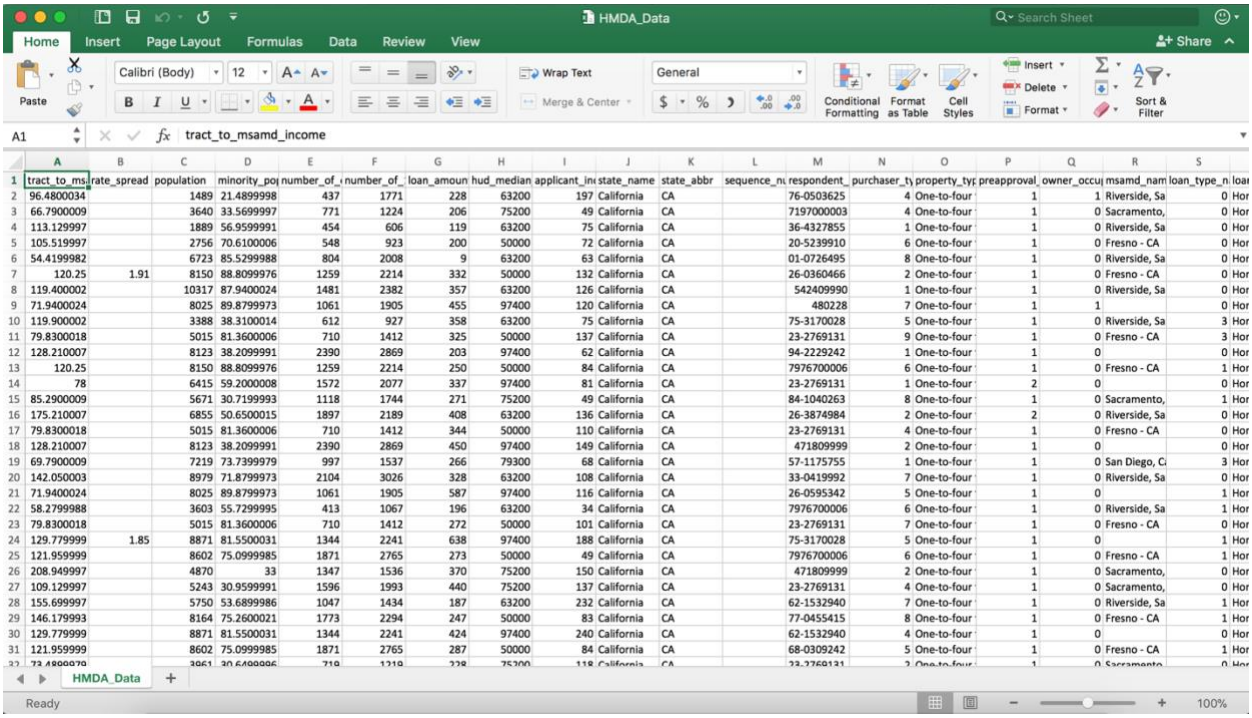
1. In the Studio, at the bottom left, click **NEW**. Then in the collection of Microsoft samples, select **Blank Experiment**.
2. Change the title of your experiment from "Experiment created on today's date" to "**YOUR PROJECT NAME**".



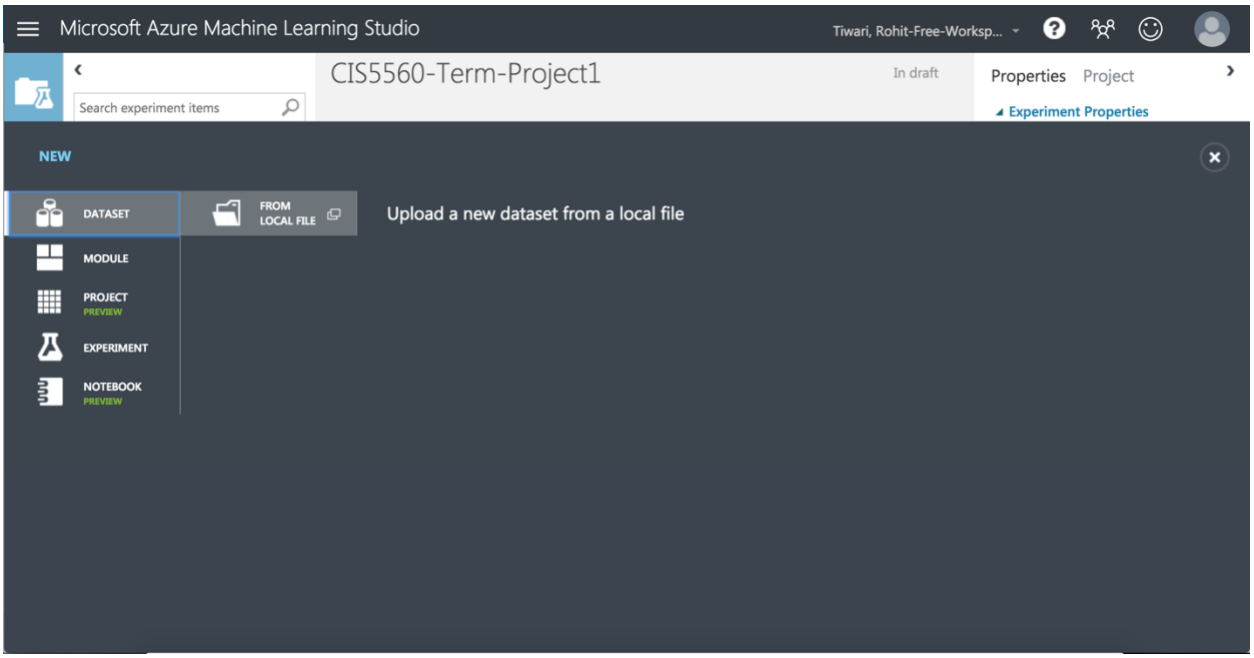
Uploading a Data File to Azure ML

When you need to create, an experiment based on your own data, or data you have obtained from a third-party, you must begin by uploading the data to Azure ML. To predict the application status of HMDA data, the dataset must be uploaded.

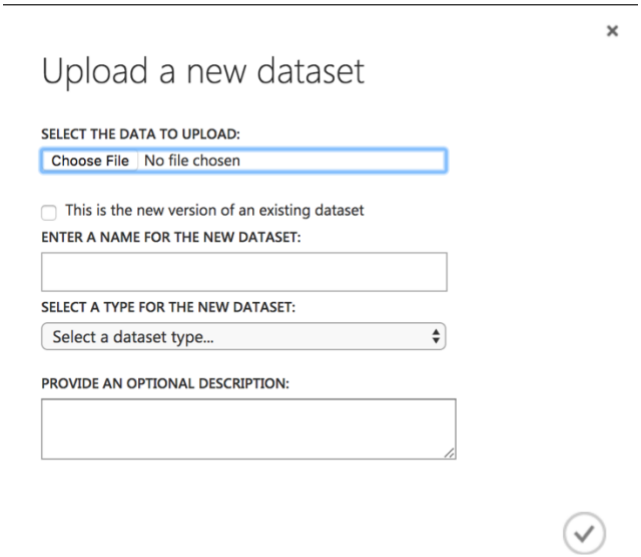
1. Open the **Restaurant.csv** file in the folder where you extracted the lab files, using either a spreadsheet application such as Microsoft Excel, or a text editor such as Microsoft Windows Notepad
2. View the contents of the file which contains all the necessary data for the prediction of the home mortgage application.



3. With your experiment open, at the bottom left, click NEW. Then in the NEW dialog box, click the DATASET tab as shown in the following image.



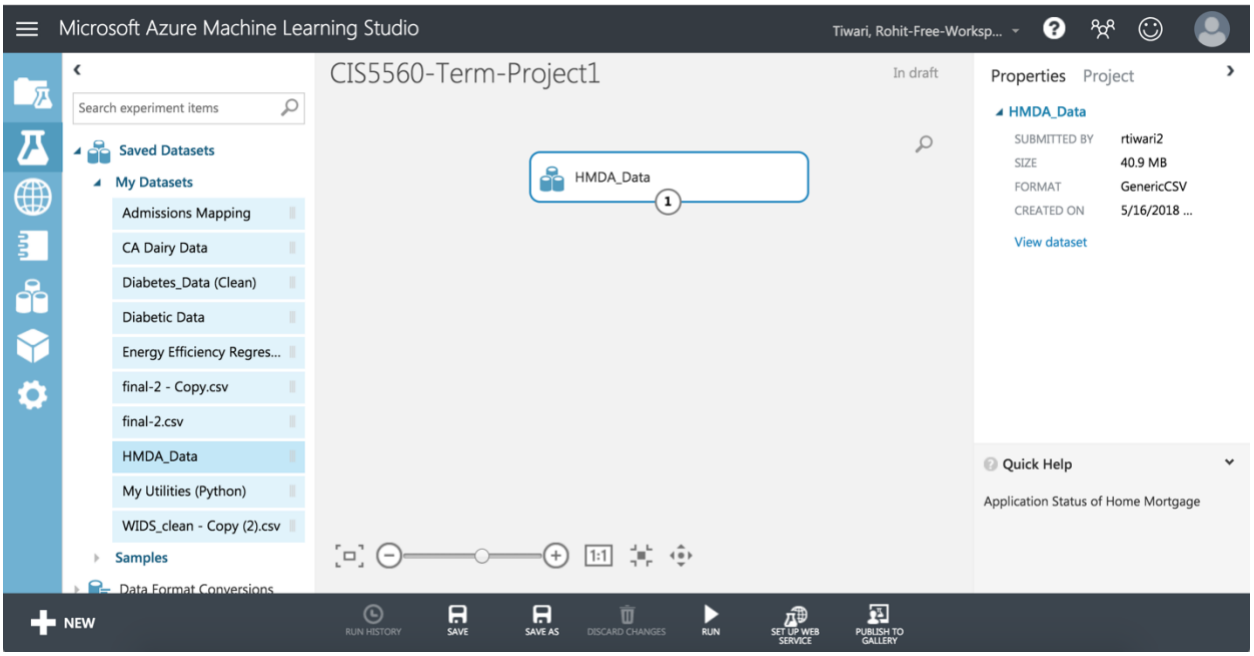
4. Click **FROM LOCAL FILE**. Then in the Upload a new dataset dialog box, browse to select the **HMDA_Data.csv** file from the folder where you extracted the lab files on your local computer and enter the following details as shown in the image below, and then click the OK icon.
   - **This is a latest version of an existing dataset**: Unselected
   - **Enter a name for the new dataset**: HMDA_Data
   - **Select a type for the new dataset**: Generic CSV file with a header (.csv)
   - **Provide an optional description**: Application Status of Home Mortgage



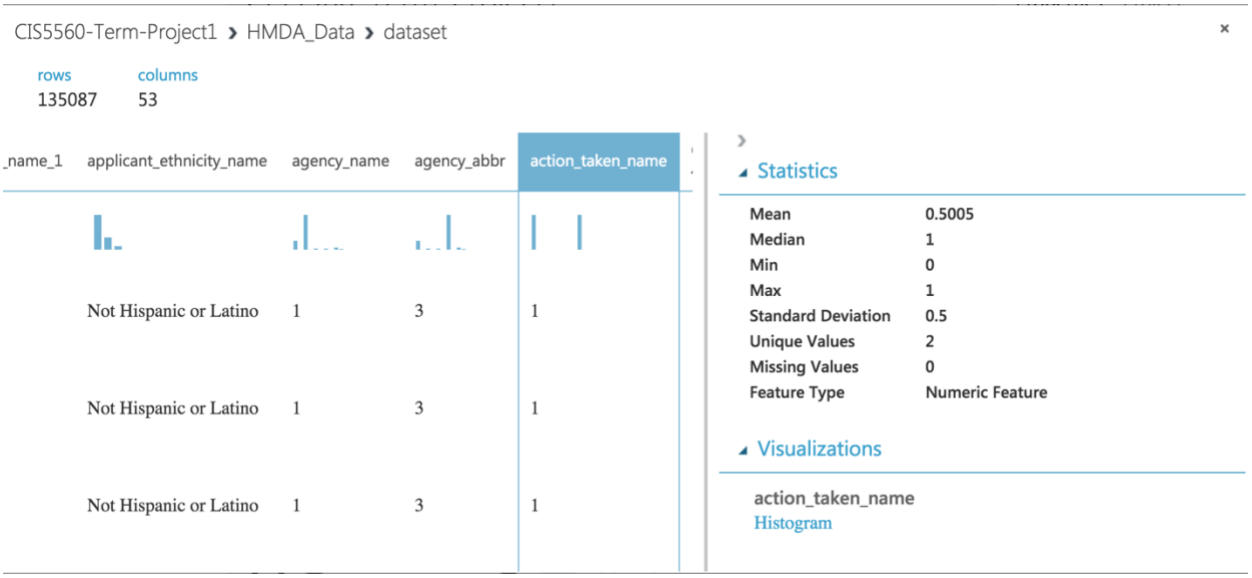
5. Wait for the upload of the dataset to be completed, and then on the experiment items pane, expand **Saved Datasets** and **My Datasets** to verify that the **HMDA_Data** dataset is listed.

Visualize the Dataset in Azure ML

1. Drag the **HMDA_Data** dataset to the canvas of your experiment.



2. Right-click the output port for the **HMDA_Data** dataset on the canvas and click **Visualize** to view the data in the dataset.
3. Verify that the dataset contains the data you viewed in the source file, and then close the dataset.



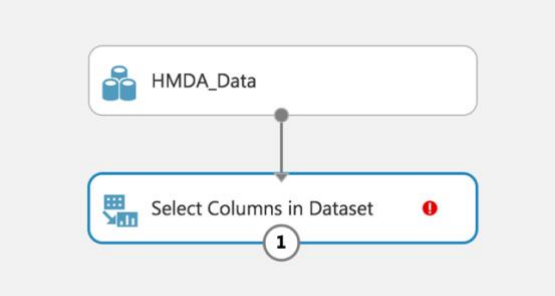
Building Two-class Logistic Regression Model

The HMDA_Data consists of various columns from which only selected columns are required for creating a model.

1. Make sure the **HMDA_Data** dataset module is already present in the module.

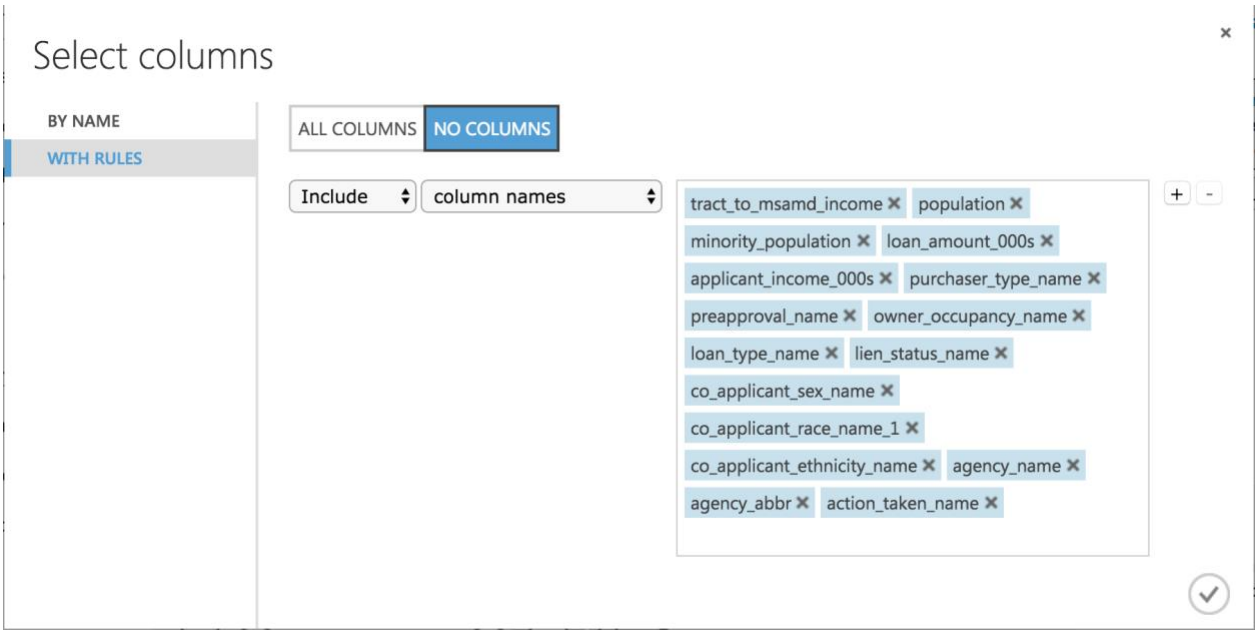
2. Search for the **Select Columns in Dataset (Project Columns)** module and drag it onto your canvas and place it under the **HMDA_Data** dataset.

3. Click the output port of the **HMDA_Data** dataset, and drag it to the input port at the top of the **Select Columns in Dataset (Project Columns)** module to connect the items. Your experiment should now look like the following image.



4. Select the **Select Columns in Dataset** module, and in the **Properties** pane on the right, click **Launch column selector**. The column selector is a common user interface element in Azure ML modules, and enables you to select the columns you want to use in the module. In this case, the **Select Columns in Dataset** module is used to filter out columns you don't need, so that only the columns you want to use are passed (or *projected*) into the data flow for the next module.

5. In the **Select columns** dialog box, select option **With Rules** to begin with **no columns**, and **include** the column names as shown in the image below. Then click the **OK** icon to close the column selector.

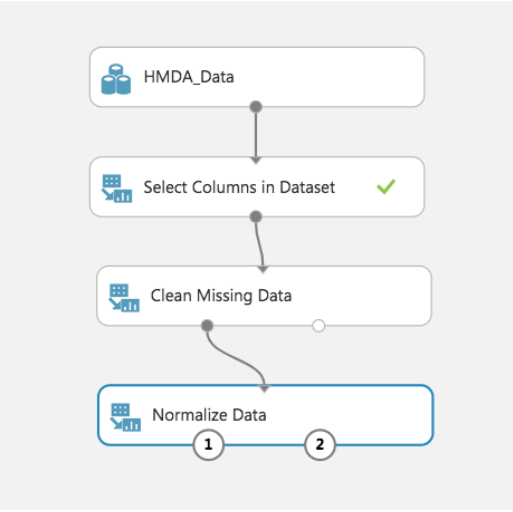


6. On the toolbar at the bottom of the page, click **SAVE** to save the experiment. Then click **RUN** to run the experiment.

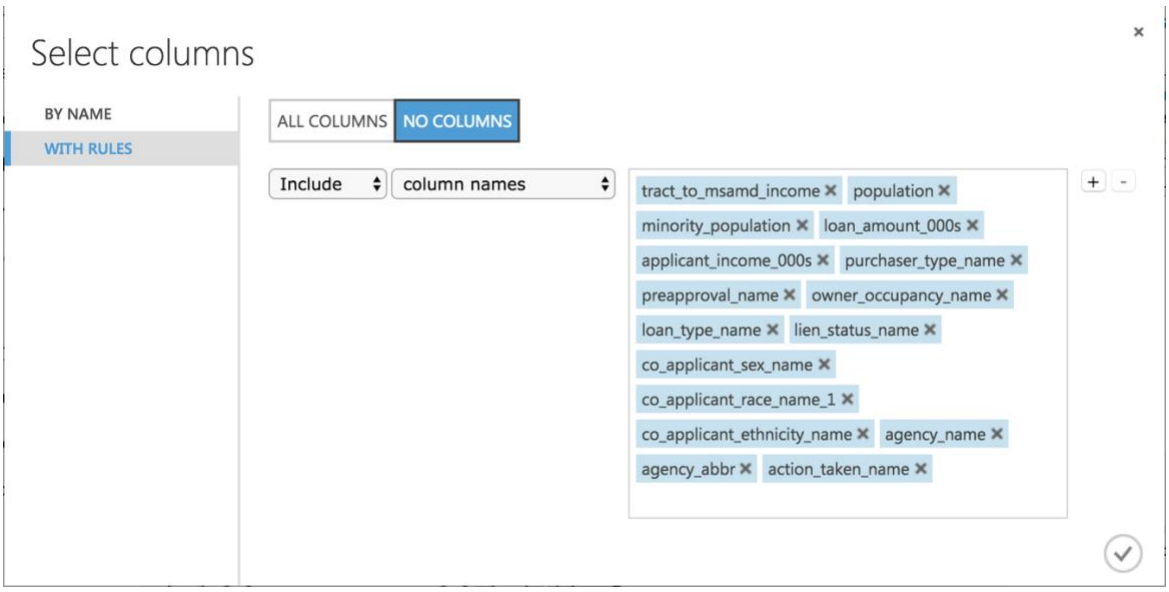
7. Search for the **Clean Missing Data** module. Drag this module onto your experiment canvas. Connect the output port of the Select Columns in Dataset (Project Columns) module to the Dataset input port of the **Clean Missing Data** module. Set the Properties of the **Clean Missing Data** module as follows:

   Cleaning mode: Remove entire row

8. Search for the **Normalize Data** module. Drag this module onto the canvas and place it under **Clean Missing Data** module. Connect the **Cleaned Dataset (Dataset)** output port from the **Clean Missing Data** module to the input port of **Normalize Data** Module. At this point your experiment should look like the following figure:



9. Search for the **Select Columns in Dataset (Project Columns)** module and drag it onto your canvas and place it under the **Normalize Data** module. Click the **Transformed dataset (dataset)** output port of the **Normalize Data** module, and drag it to the input port at the top of the **Select Columns in Dataset (Project Columns)** module to connect the items. Select the **Select Columns in Dataset** module, and in the **Properties** pane on the right, click **Launch column selector.** In the **Select columns** dialog box, select option **With Rules** to begin with **no columns**, and **include** the column names as shown in the image below. Then click the **OK** icon to close the column selector.

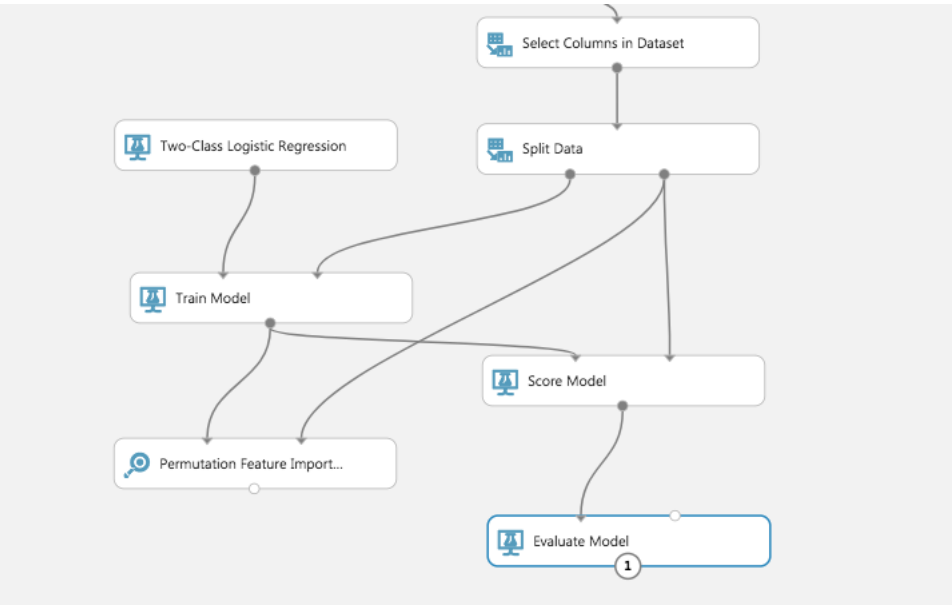


10. Search for the **Split Data (Split)** module. Drag this module onto your experiment canvas. Connect the **Results dataset** output port of the **Select Columns in Dataset (Project Columns)** module to the **Dataset** input port of the **Split** module. Set the **Properties** of the **Split** module as follows:
    - **Splitting mode:** Split Rows
    - **Randomized split:** Checked
    - **Random seed:** 4567
    - **Stratified split:** False
    - **Fraction of rows in the first output:** 0.7

11. Search for the **Two-Class Logistic Regression** module. Drag this module onto the canvas. Set the properties of this module as follows:
    Memory size for L-BFGS: 10
    Random number seed: 4567

12. Search for the **Train Model** module. Drag this module onto the canvas.
13. Connect the **Untrained Model** output port of the **Two-class Logistic Regression** module to the **Untrained Model** input port of the **Train Model** module.
14. Connect the **Results dataset1** (left) output port of the **Split** module to the **Dataset input** port of the **Train model** module.

15. Select the **Train Model** module. Then, on the **Properties** pane, launch the column selector and select the **action_taken_name** column: Include > column names > action_taken_name.
16. Search for the **Score Model** module and drag it onto the canvas.
17. Connect the **Trained Model** output port of the of the **Train Model** module to the **Trained Model** input port of the **Score Model** module. Then connect the **Results dataset2** (right) output port of the **Split** module to the **Dataset** port of the **Score Model** module.
18. Search for the **Permutation Feature Importance** module and drag it onto the canvas.
19. Connect the **Trained Model** output port of the **Train Model** module to the **Trained model** input port of the **Permutation Feature Importance** module. Then connect the **Results dataset2** (right) output port of the **Split** module to the **Dataset** port of the **Test data** input port of the Permutation Feature Importance module.
20. Select the **Permutation Feature Importance** module and in the **Properties** pane set the following parameters:
    - **Random Seed:** 4567
    - **Metric for measuring performance:** Classification – Accuracy
21. Search for the **Evaluate Model** module and drag it onto the canvas. Connect the **Scored Dataset** output port of the **Score Model** module to the left hand **Scored dataset** input port of the **Evaluate Model** module. The new portion of your experiment should now look like the following:



Sweep Model Parameters

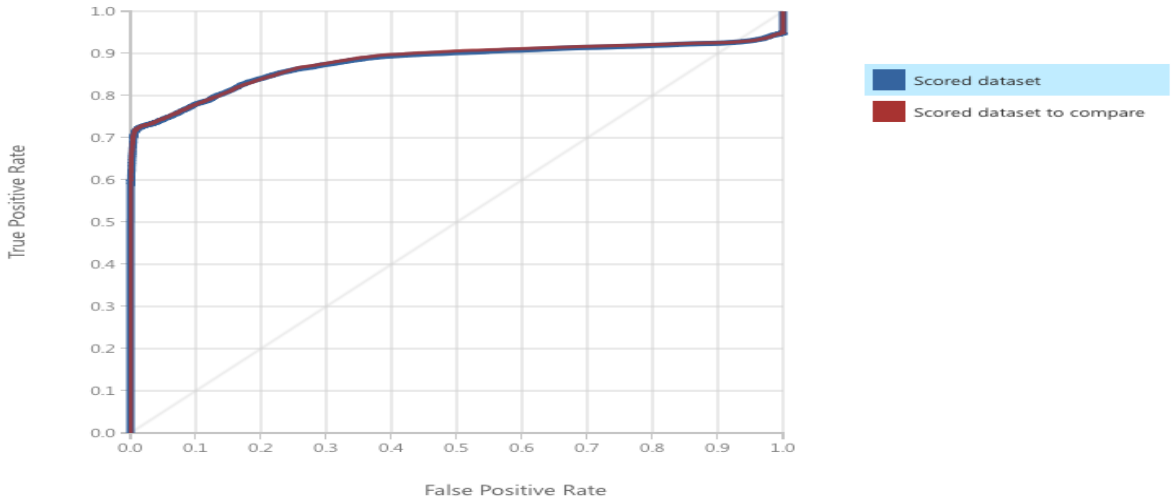
You will now improve the machine learning model by sweeping the parameter space.

22. Select all of the modules below the **Select Columns in Dataset (Project Columns)** module you added at step 9. Then copy and paste these modules onto the canvas and drag the copies to one side.
23. Connect the **Results dataset** output of the **Select Columns in Dataset (Project Columns)** module to the **Dataset input** of the new **Split** module.

24. Remove the copied **Permutation Feature Importance**, **Train Model**, and **Evaluate Model** modules.
25. Search for the **Tune Model Hyperparameters (Sweep Parameters)** module. Drag this module onto the canvas in place of the **Train Model** module you removed.
26. Connect the **Untrained model** output port of the new **Two-class Logistic Regression** module to the **Untrained model** (left) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
27. Connect the **Results dataset1** (left) output port of the new **Split** module to the **Training dataset** (middle) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
28. Connect the **Results dataset2** (right) output port of the **Split** module to the **Optional test dataset** (right) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
29. Connect the **Trained model** (right) output of the **Tune Model Hyperparameters (Sweep Parameters)** module to the **Trained model** (left) input of the new **Score Model** module.
30. Click the **Tune Model Hyperparameters (Sweep Parameters)** module to expose the **Properties** pane. Set the properties as follows so that 20 combinations of parameters are randomly tested:
    - **Specify parameter sweeping mode:** Random sweep
    - **Random seed:** 4567
    - **Column Selector:** action_taken_name
    - **Metric for measuring performance for regression:** Coefficient of determination
    - **Maximum number of runs on random sweep:** 20
    - **Metric for measuring performance for classification:** Accuracy

31. Connect the **Scored dataset** output port of the copied **Score Model** module to the **Scored dataset to compare** (right) input port of the original **Evaluate Model** module for the first **Two-class Logistic Regression** model.

32. Save and run the experiment. When the experiment is finished, visualize the **Evaluation results** output port of the **Evaluate Model** module.



ROC curve is 0.878

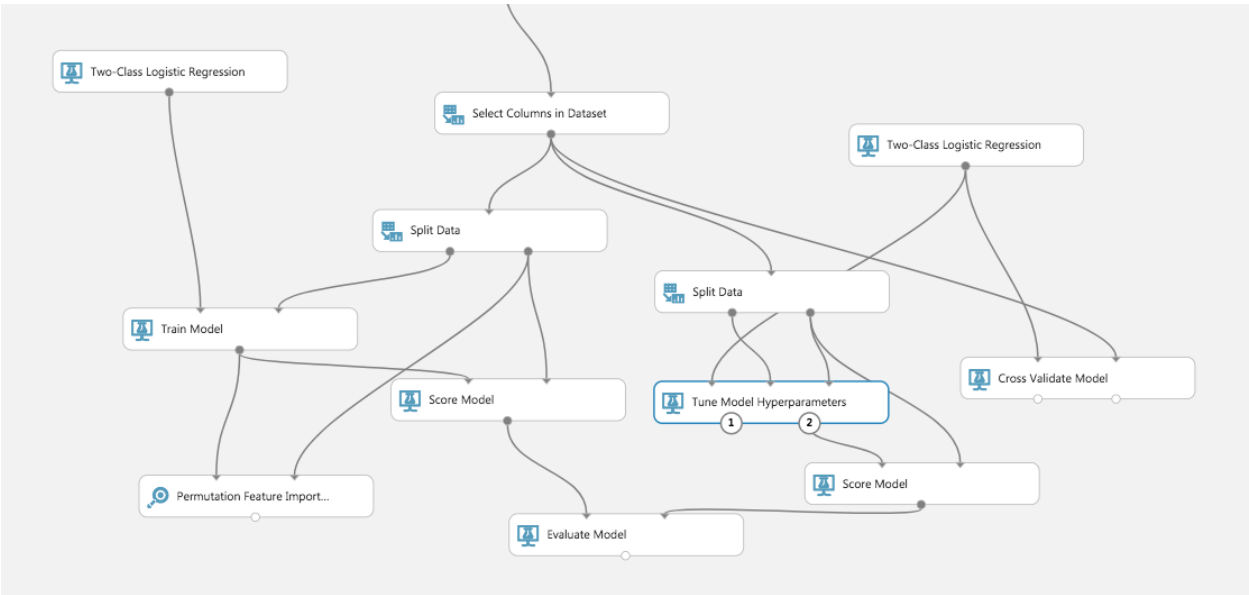
Cross Validate the Model

Cross validation or a machine learning model uses resampling of the dataset to test the performance on a number of training and testing data subsets. Each training and testing data subset sampled from the complete data set is called a fold. Ideally, a good machine learning model should work well regardless of the test data used. When cross validated, a good model will have similar performance across the folds. This property of good machine learning models is known as generalization. A model which generalizes produces good results for any possible set of valid input values. Models that generalize can be expected to work well in production.

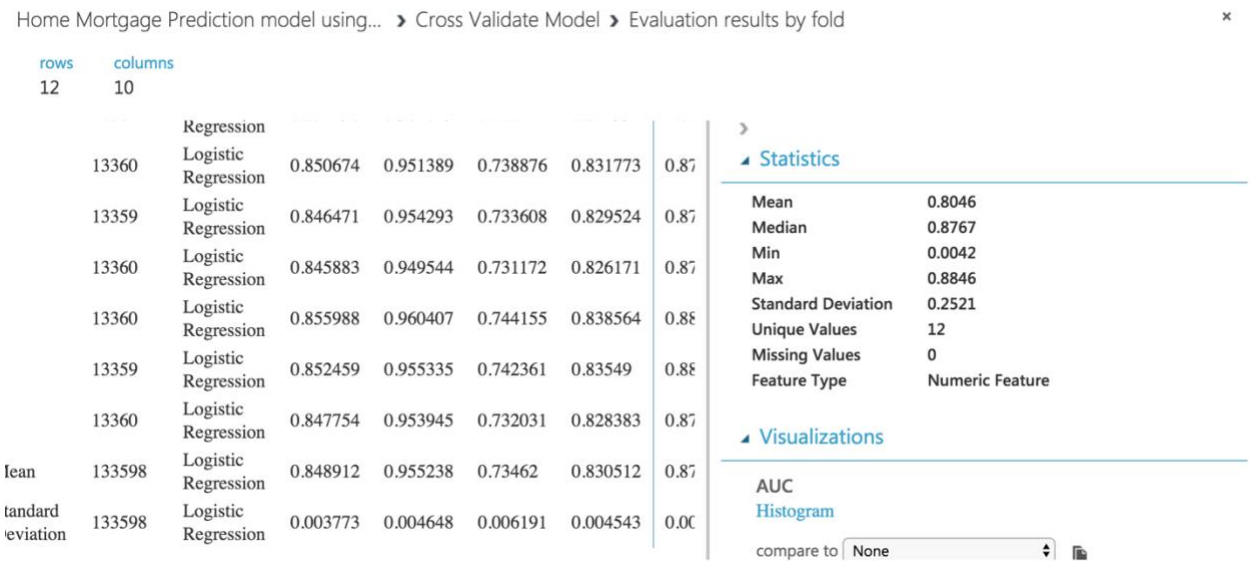
33. Search for the **Cross Validate Model** module. Drag this module onto the canvas.
34. Connect the **Untrained model** output from the most recently added **two-class logistic regression Model** module (the one connected to the **tune model hyperparameters** module) to the **Untrained model** input port of the **Cross Validate Model** module.
35. Connect the **Results dataset** output port of the **Select Columns in Dataset (Project Columns)** module to the **Dataset input** port of the **Cross Validate Model** module.

36. Select the **Cross Validate Model** module, and set its properties as follows:
    - **Column Selector**: action_taken_name
    - **Random seed**: 3467

Your experiment should resemble the following:



37. Save and run the experiment. When the experiment has finished, visualize the **Evaluation Results by Fold** (right) output port of the **Cross Validate Model** module.

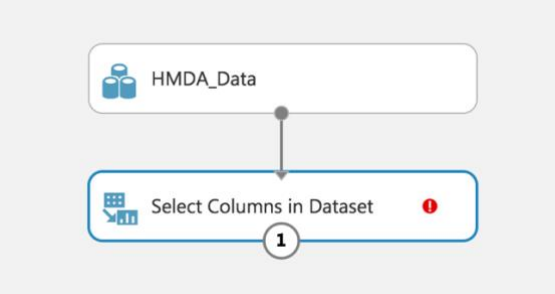


Building Two-class Decision Forest Model:

1. Make sure the **HMDA_Data** dataset module is already present in the module.

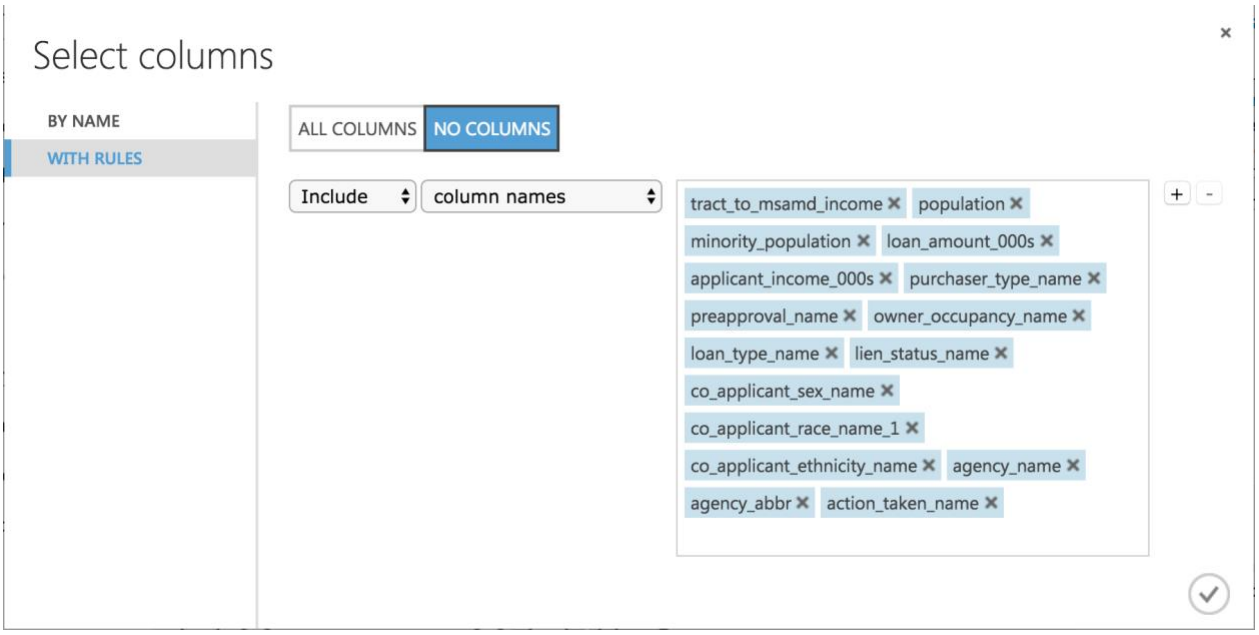
2. Search for the **Select Columns in Dataset (Project Columns)** module and drag it onto your canvas and place it under the **HMDA_Data** dataset.

3. Click the output port of the **HMDA_Data** dataset, and drag it to the input port at the top of the **Select Columns in Dataset (Project Columns)** module to connect the items. Your experiment should now look like the following image.



4. Select the **Select Columns in Dataset** module, and in the **Properties** pane on the right, click **Launch column selector**. The column selector is a common user interface element in Azure ML modules, and enables you to select the columns you want to use in the module. In this case, the **Select Columns in Dataset** module is used to filter out columns you don't need, so that only the columns you want to use are passed (or *projected*) into the data flow for the next module.

5. In the **Select columns** dialog box, select option **With Rules** to begin with **no columns**, and **include** the column names as shown in the image below. Then click the **OK** icon to close the column selector.

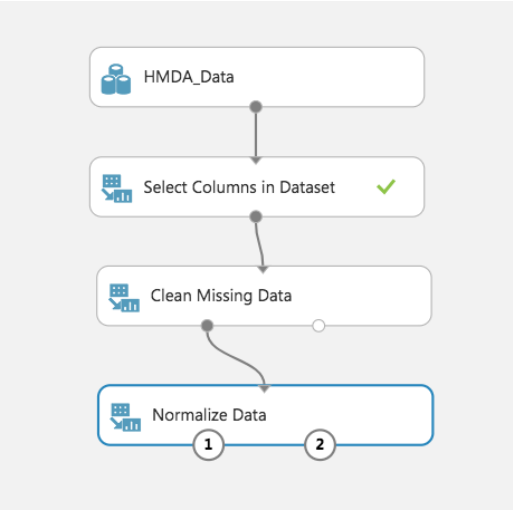


6. On the toolbar at the bottom of the page, click **SAVE** to save the experiment. Then click **RUN** to run the experiment.

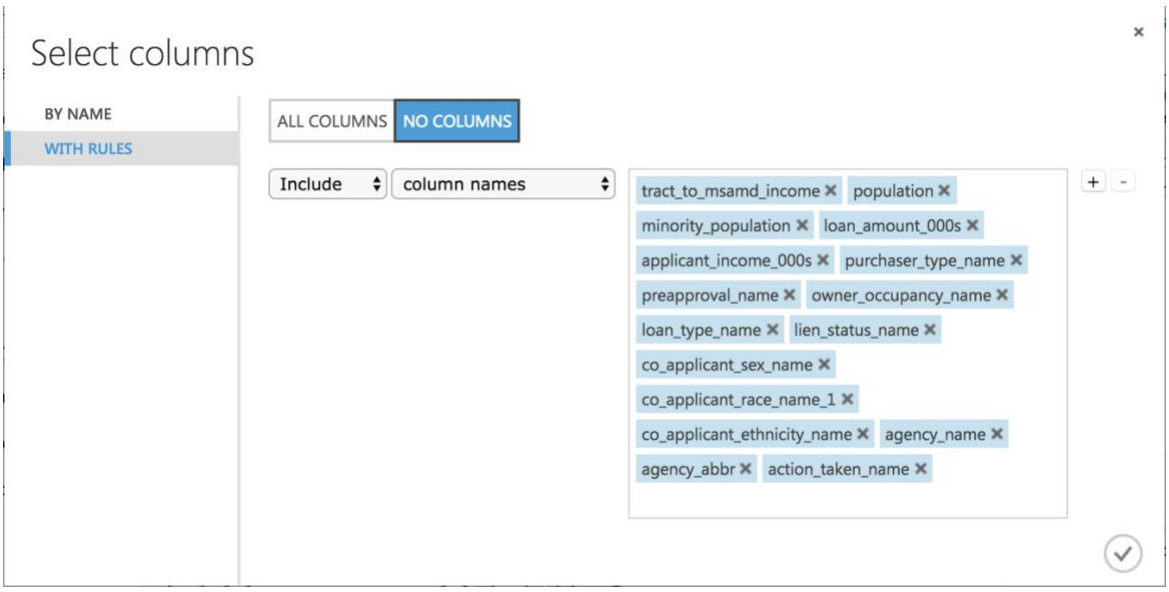
7. Search for the **Clean Missing Data** module. Drag this module onto your experiment canvas. Connect the output port of the Select Columns in Dataset (Project Columns) module to the Dataset input port of the **Clean Missing Data** module. Set the Properties of the **Clean Missing Data** module as follows:

   Cleaning mode: Remove entire row

8. Search for the **Normalize Data** module. Drag this module onto the canvas and place it under **Clean Missing Data** module. Connect the **Cleaned Dataset (Dataset)** output port from the **Clean Missing Data** module to the input port of **Normalize Data** Module. At this point your experiment should look like the following figure:



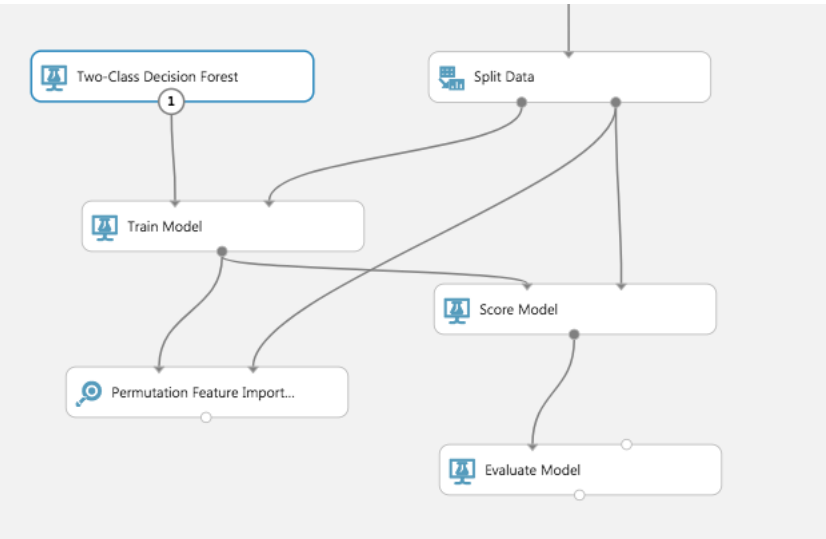
9. Search for the **Select Columns in Dataset (Project Columns)** module and drag it onto your canvas and place it under the **Normalize Data** module. Click the **Transformed dataset (dataset)** output port of the **Normalize Data** module, and drag it to the input port at the top of the **Select Columns in Dataset (Project Columns)** module to connect the items. Select the **Select Columns in Dataset** module, and in the **Properties** pane on the right, click **Launch column selector.** In the **Select columns** dialog box, select option **With Rules** to begin with **no columns**, and **include** the column names as shown in the image below. Then click the **OK** icon to close the column selector.



10. Search for the **Split Data (Split)** module. Drag this module onto your experiment canvas. Connect the **Results dataset** output port of the **Select Columns in Dataset (Project Columns)** module to the **Dataset** input port of the **Split** module. Set the **Properties** of the **Split** module as follows:
    - **Splitting mode:** Split Rows
    - **Randomized split:** Checked
    - **Random seed:** 4567
    - **Stratified split:** False
    - **Fraction of rows in the first output:** 0.7

11. Search for the **Two Class Decision Forest** module. Make sure you have selected the regression model version of this algorithm. Drag this module onto the canvas. Set the Properties if this module as follows:
    - **Resampling method:** Bagging
    - **Create trainer mode:** Single Parameter
    - **Number of decision trees:** 40
    - **Maximum depth of the decision trees:** 32
    - **Number of random Splits per node:** 256
    - **Minimum number of samples per leaf node:** 4567
12. Search for the **Train Model** module. Drag this module onto the canvas.

13. Connect the **Untrained Model** output port of the **Two-class Decision Forest** module to the **Untrained Model** input port of the **Train Model** module.
14. Connect the **Results dataset1 (left)** output port of the **Split module** to the Dataset input port of the **Train model** module.
15. Select the **Train Model** module. Then, on the Properties pane, launch the column selector and select the **action_taken_name** column: Include > column names > action_taken_name.
16. Search for the **Score Model** module and drag it onto the canvas.
17. Connect the **Trained Model** output port of the of the **Train Model** module to the **Trained Model** input port of the **Score Model** module. Then connect the **Results dataset2 (right)** output port of the **Split module** to the Dataset port of the **Score Model** module.
18. Search for the **Permutation Feature Importance** module and drag it onto the canvas.
19. Connect the **Trained Model** output port of the **Train Model** module to the **Trained model** input port of the **Permutation Feature Importance** module. Then connect the **Results dataset2 (right**) output port of the **Split module** to the **Dataset port** of the **Test data** input port of the **Permutation Feature Importance** module.
20. Select the **Permutation Feature Importance** module and in the Properties pane set the following parameters:
    - **Random Seed:** 4567
    - **Metric for measuring performance:** Classification – Accuracy
21. Search for the **Evaluate Model** module and drag it onto the canvas. Connect the **Scored Dataset** output port of the **Score Model** module to the left hand **Scored dataset** input port of the **Evaluate Model** module. The new portion of your experiment should now look like the following:



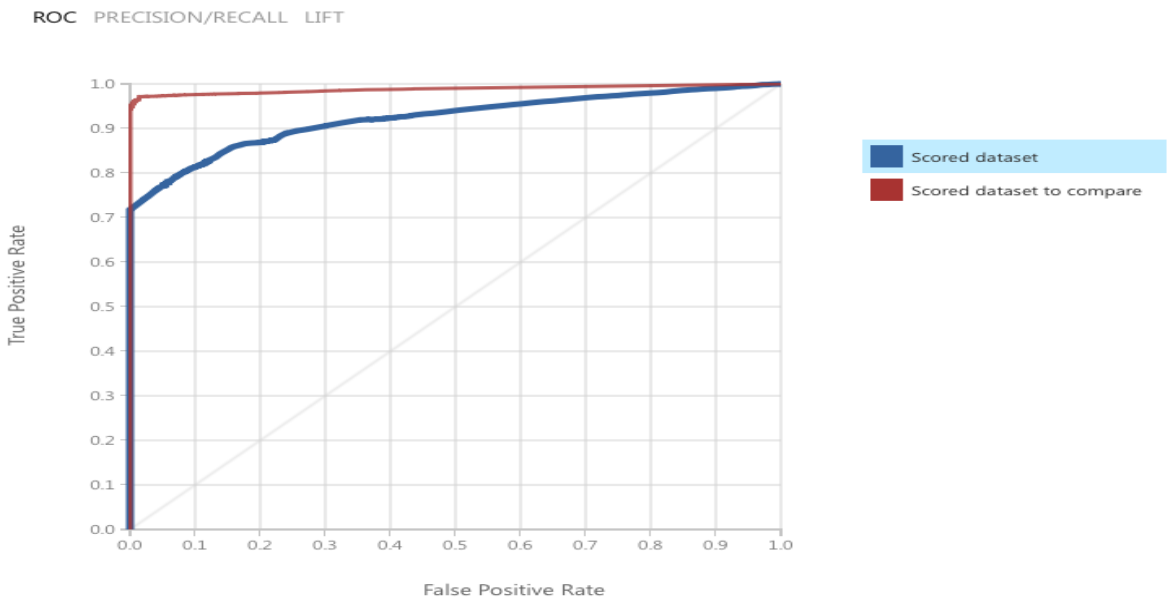
Sweep Model Parameters

You will now improve the machine learning model by sweeping the parameter space.

22. Select all of the modules below the **Select Columns in Dataset (Project Columns)** module you added at step 9. Then copy and paste these modules onto the canvas and drag the copies to one side.
23. Connect the **Results dataset** output of the **Select Columns in Dataset (Project Columns)** module to the **Dataset input** of the new **Split** module.
24. Remove the copied **Permutation Feature Importance**, **Train Model**, and **Evaluate Model** modules.
25. Search for the **Tune Model Hyperparameters (Sweep Parameters)** module. Drag this module onto the canvas in place of the **Train Model** module you removed.
26. Connect the **Untrained model** output port of the new **Two-class Decision Forest Model** module to the **Untrained model** (left) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
27. Connect the **Results dataset1** (left) output port of the new **Split** module to the **Training dataset** (middle) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
28. Connect the **Results dataset2** (right) output port of the **Split** module to the **Optional test dataset** (right) input port of the **Tune Model Hyperparameters (Sweep Parameters)** module.
29. Connect the **Trained model** (right) output of the **Tune Model Hyperparameters (Sweep Parameters)** module to the **Trained model** (left) input of the new **Score Model** module.
30. Click the **Tune Model Hyperparameters (Sweep Parameters)** module to expose the **Properties** pane. Set the properties as follows so that 5 combinations of parameters are randomly tested:
    - **Specify parameter sweeping mode:** Random sweep
    - **Random seed:** 2345
    - **Column Selector:** action_taken_name
    - **Metric for measuring performance for regression:** Root of mean squared error
    - **Maximum number of runs on random sweep:** 5
    - **Metric for measuring performance for classification:** Accuracy

31. Connect the **Scored dataset** output port of the copied **Score Model** module to the **Scored dataset to compare** (right) input port of the original **Evaluate Model** module for the first **Two-class Logistic Regression** model.

32. Save and run the experiment. When the experiment is finished, visualize the **Evaluation results** output port of the **Evaluate Model** module.



ROC curve is 0.922

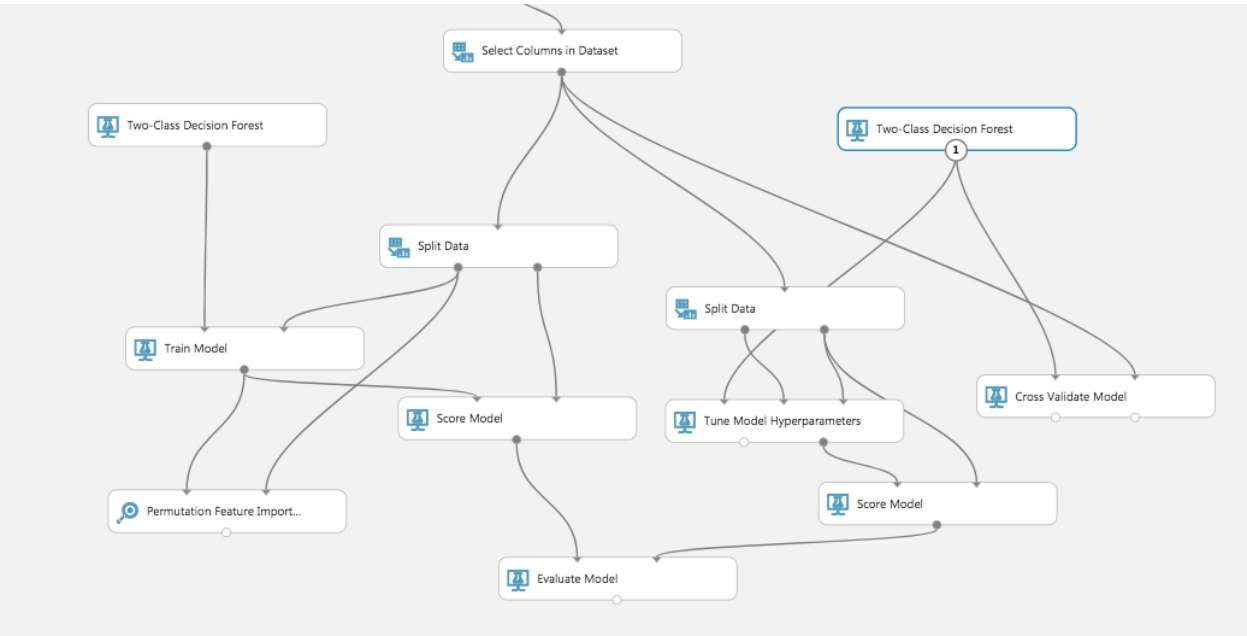
Cross Validate the Model

Cross validation or a machine learning model uses resampling of the dataset to test the performance on a number of training and testing data subsets. Each training and testing data subset sampled from the complete data set is called a fold. Ideally, a good machine learning model should work well regardless of the test data used. When cross validated, a good model will have similar performance across the folds. This property of good machine learning models is known as generalization. A model which generalizes produces good results for any possible set of valid input values. Models that generalize can be expected to work well in production.

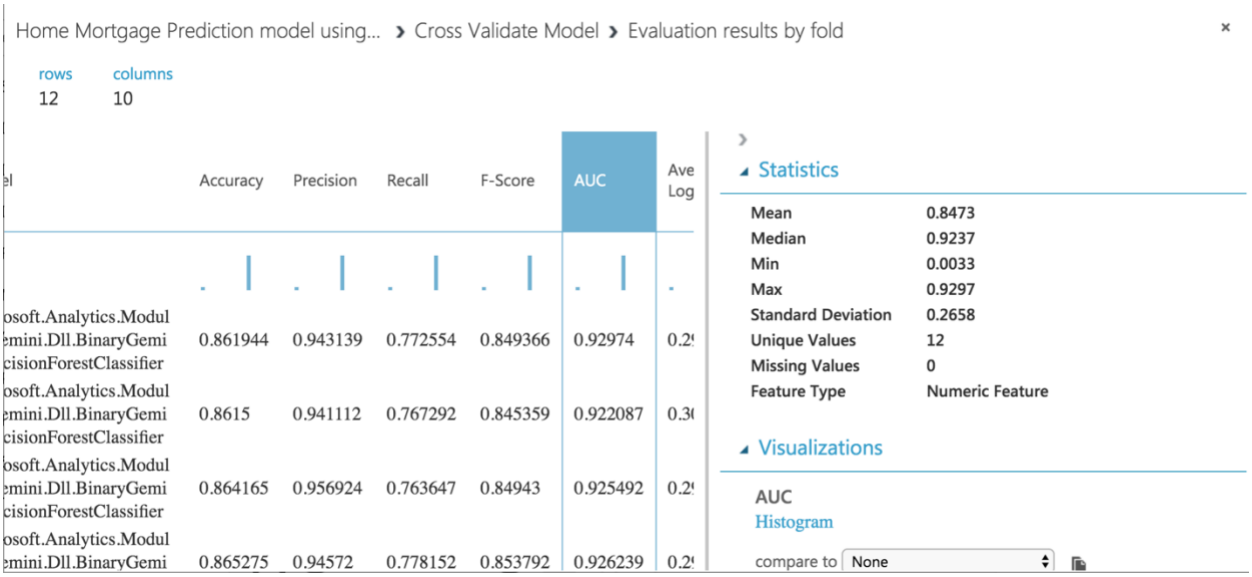
33. Search for the **Cross Validate Model** module. Drag this module onto the canvas.
34. Connect the **Untrained model** output from the most recently added **two-class logistic regression Model** module (the one connected to the **tune model hyperparameters** module) to the **Untrained model** input port of the **Cross Validate Model** module.
35. Connect the **Results dataset** output port of the **Select Columns in Dataset (Project Columns)** module to the **Dataset input** port of the **Cross Validate Model** module.

36. Select the **Cross Validate Model** module, and set its properties as follows:
    - **Column Selector**: action_taken_name
    - **Random seed**: 3467

Your experiment should resemble the following:



37. Save and run the experiment. When the experiment has finished, visualize the **Evaluation Results by Fold** (right) output port of the **Cross Validate Model** module.



Summary

Two Class Logistic Regression	Two-class Decision Forest
AUC: 0.87	AUC: 0.92

Reference:

- **Dataset Source - https://www.consumerfinance.gov/data-research/hmda/explore**
- **Azure ML Studio - https://studio.azureml.net/**
- **Azure Microsoft Docs - https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/machine-learning-studio-algorithm-and-module-help**