CIS5560 Term Project Tutorial

Azure ML Lab Tutorial

On

Predictive Analysis Of New York Motor Vehicles Collisions

05/19/2017

Submitted by:

Anusha Manjappa(amanjap@calstatela.edu)

Prathibha Gubbi Prakash(pgubbip@calstate.edu)

Srihitha Reddy Sivannagari(ssivann@calstatela.edu)

Instructor:

Prof. Jongwook Woo

Objective

The main objective of this tutorial is to perform predictive analysis on the Borough of New York Motor Vehicles collisions data set, based on the Vehicle type code, Number of persons killed and Contributing factor using Microsoft Azure Machine Learning Studio.

Step 1: Creating an Azure ML Experiment

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

Step 2: 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.

Step 3: Sign up for a free Azure ML

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.

Step 4: Creating an Azure ML Experiment for Classification

1. Azure ML enables us to create experiments in which we can manipulate data, create predictive models, and visualize the results.

2. In this tutorial, you will create a simple experiment in which you will explore a sample dataset that contains details on the numerous incidents of, **Motor Vehicle Collisions In New York City** from which you would like to predict score of the borough (area) which had the highest number of incidents based on '**Vehicle type code**', '**Number of persons killed**' and **'Contributing factor**'.

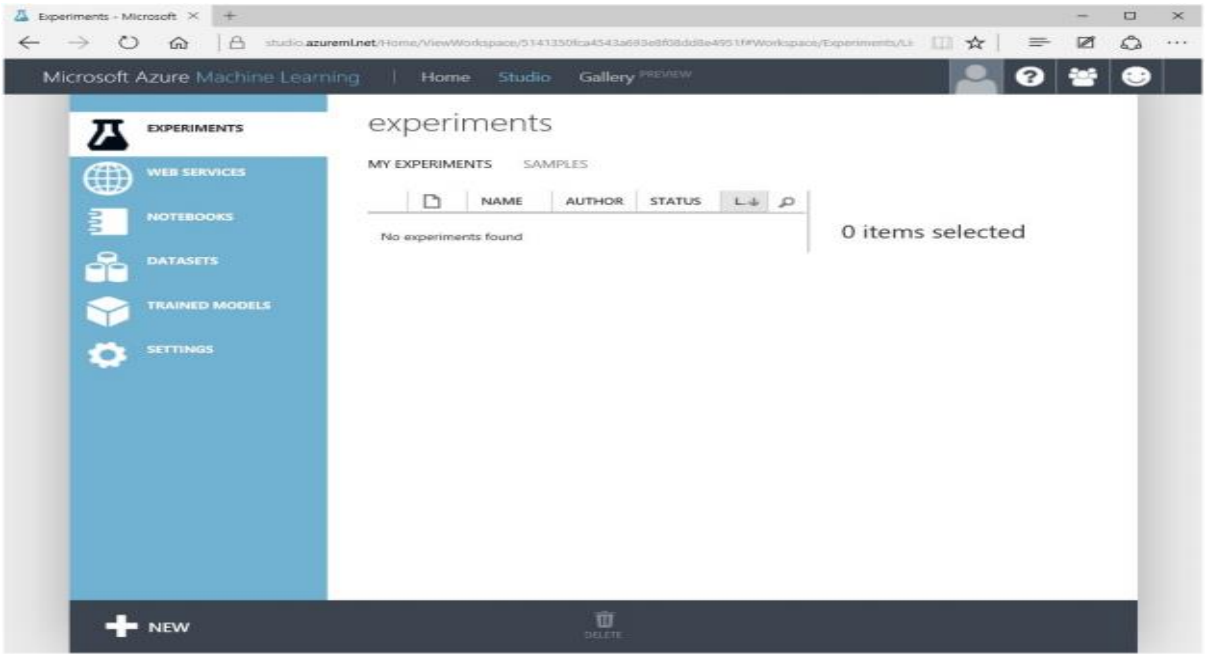
Sign into Azure ML Studio

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 Studio tab at the top of the page).



5. In the Studio, at the bottom left, click **NEW**. Then in the collection of Microsoft samples, select **Blank Experiment**.

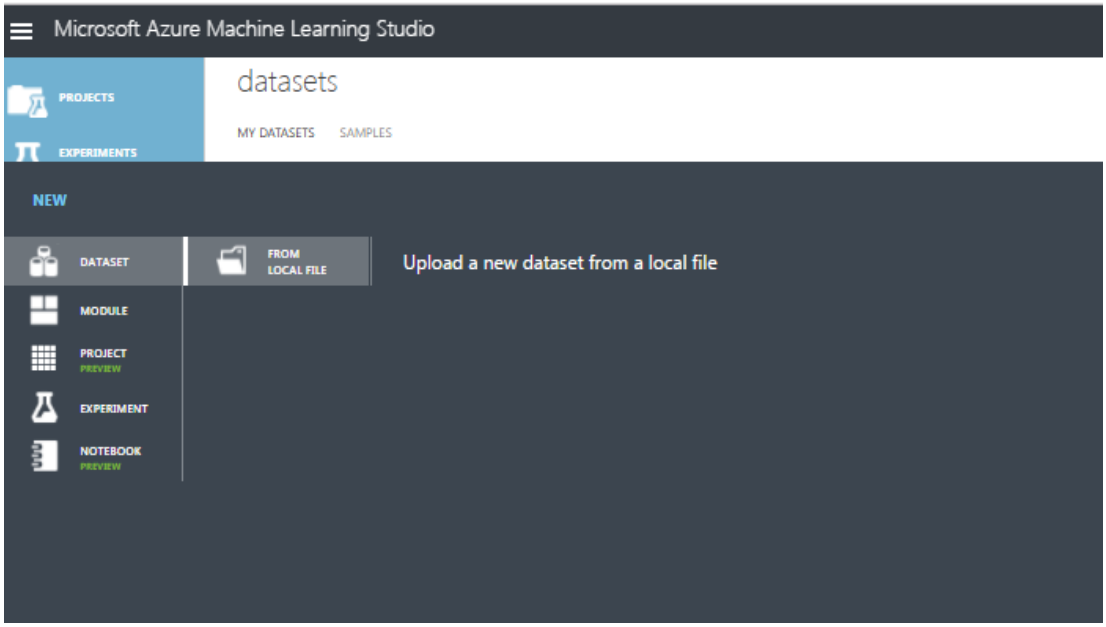
6. Change the title of your experiment from "Experiment created on today's date" to "**NYMVC**".

Uploading a Data File to Azure ML:

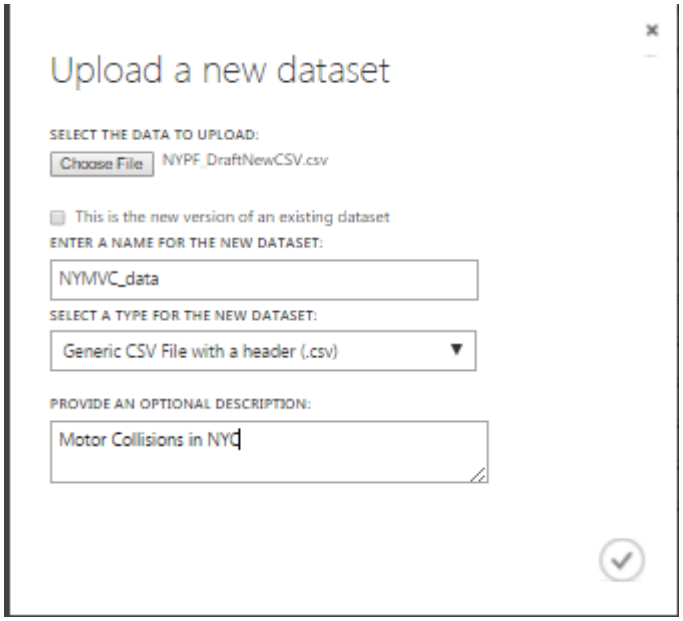
When you need to create, an experiment based on your own data or data you have obtained from a third-party, you must begin by uploading the data to Azure ML. To predict the BORUGH based on number of incidents, the dataset must be uploaded.

1. Open the **NYMVC_data.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. With the **NYMVC** experiment open, at the bottom left, click NEW. Then in the NEW dialog box, click the DATASET tab as shown in the below image.



3. Click **FROM LOCAL FILE**. Then in the Upload a new dataset dialog box, browse to select the **NYMVC_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**: **NYMVC_data**
   • **Select a type for the new dataset**: Generic CSV file with a header (.csv)
   • **Provide an optional description**: Motor Collisions in NYC



4. 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 dataset is listed.

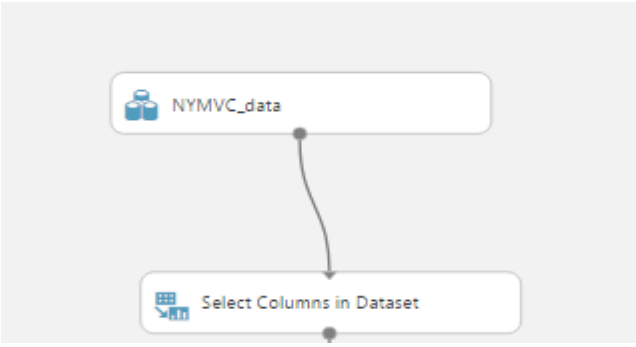
Visualize the Dataset in Azure ML:

1. Drag the **NYMVC_data.csv** dataset to the canvas for the **NYMVC** experiment.

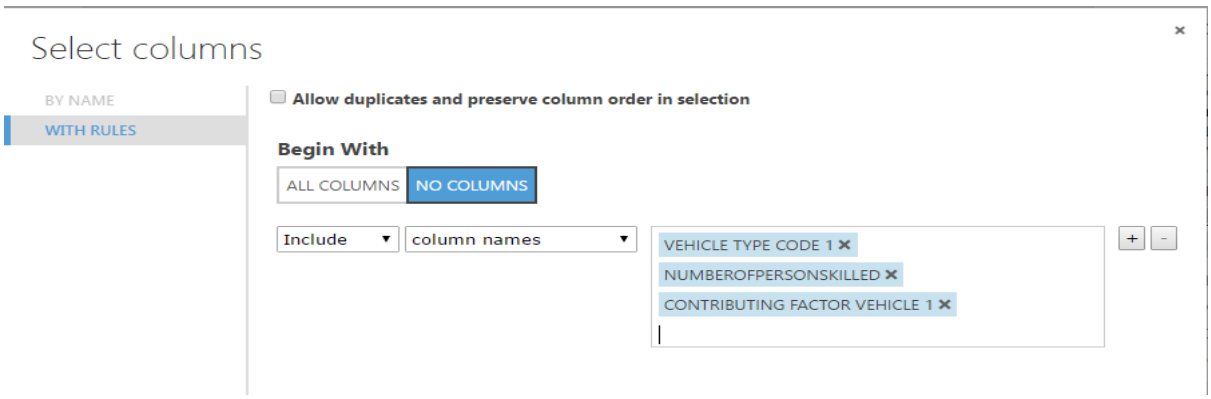
2. Right-click the output port for the **NYMVC_data.csv** dataset on the canvas and click **Visualize** to view the data in the dataset.

3. Verify that the dataset contains the data you viewed in the source file, and then close the dataset.

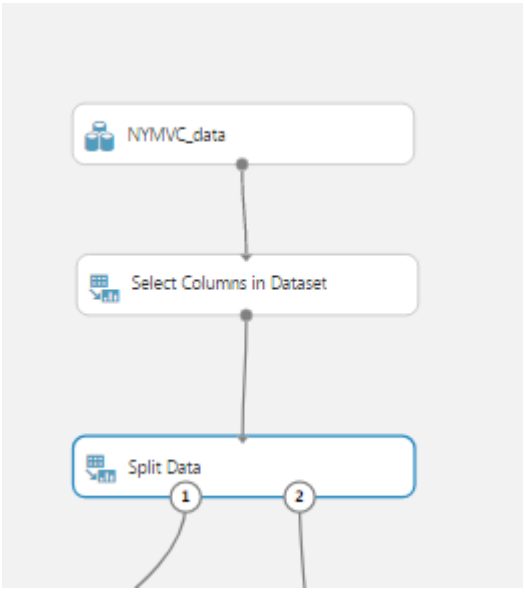
4. Search for the **Select Columns in Dataset (Project Columns)** module and drag it onto your canvas. Connect the Results Dataset output of the **NYMVC_data.csv** module to the input port of the **Select Columns in Dataset (Project Columns)** module.



5. In the properties pane, select **launch column selector** and select the with rules option. Under **no columns** include the column names **Vehicle type code**, **Number of persons killed** and **Contributing factor**.



6. 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 Data** (Split) module. Set the Properties of the **Split Data** (Split) module as follows:
    a. **Splitting mode**: Split Rows
    b. **Fraction of rows in the first output**: 0.7
    c. **Randomized Split**: Checked
    d. **Random seed**: 0
    e. **Stratified Split**: False

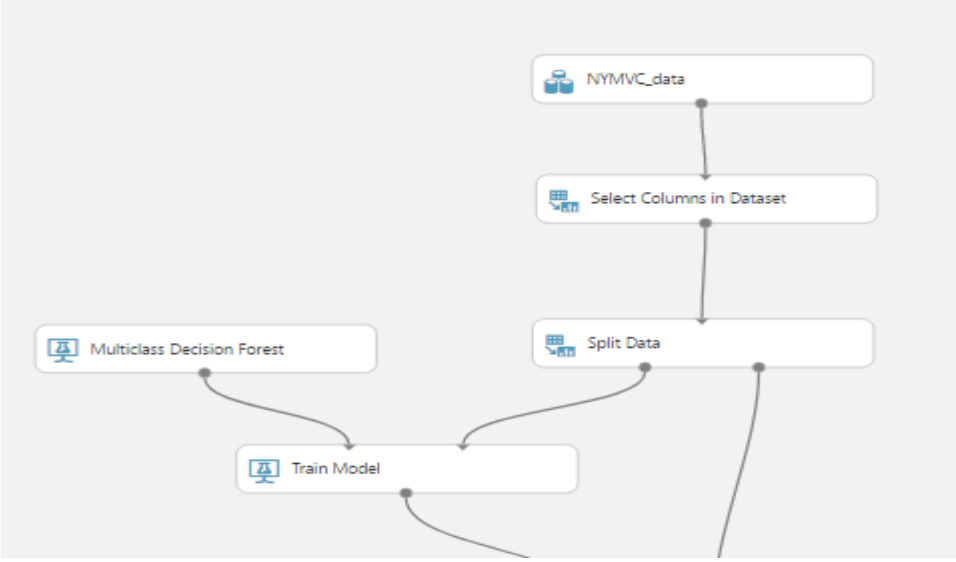


7. Search for the **Multiclass Decision Forest** module. Drag this module onto the canvas.

8. Set the Properties of this module as follows:
    a. **Resampling method**: Bagging
    b. **Create trainer mode**: Single Parameter
    c. **Number of decision trees:** 8
    d. **Maximum depth of the decision tree:** 32
    e. **Number of random split per node:** 128
    f. **Minimum number of samples per leaf nodes:** 1
    g. **Allow unknown values for categorical features:** unchecked

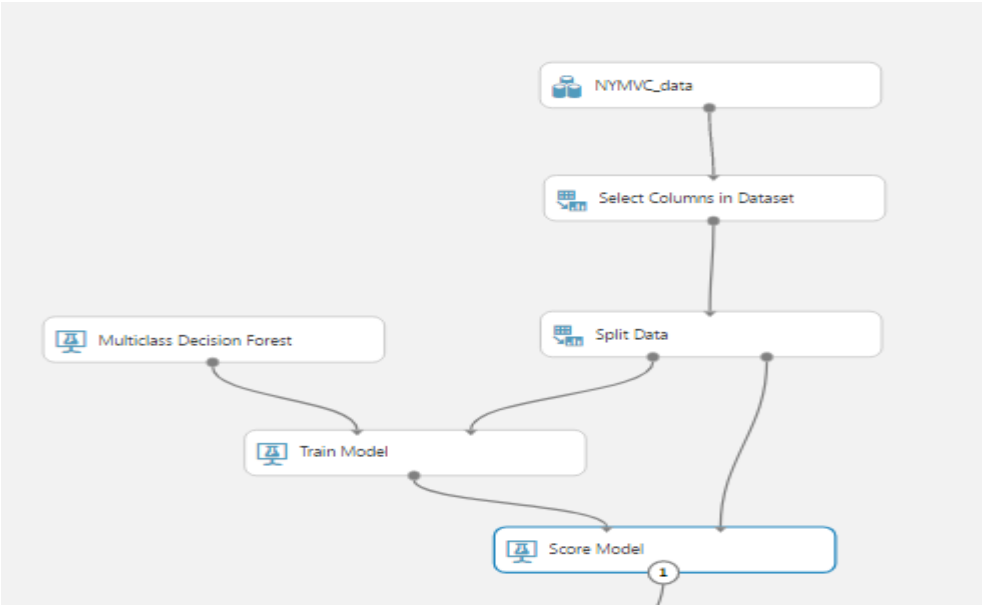
9. Search for the **Train Model** module. Drag this module onto the canvas.

10. Connect the **Untrained Model output** port of the **Multiclass Decision Forest** module to the **Untrained Model** input port of the Train Model module. Connect the **Results dataset1** output port of the **Split Data** (Split) module to the **Dataset** input port of the Train model module. On the Properties pane, launch the column selector and select the **SCORE** column.

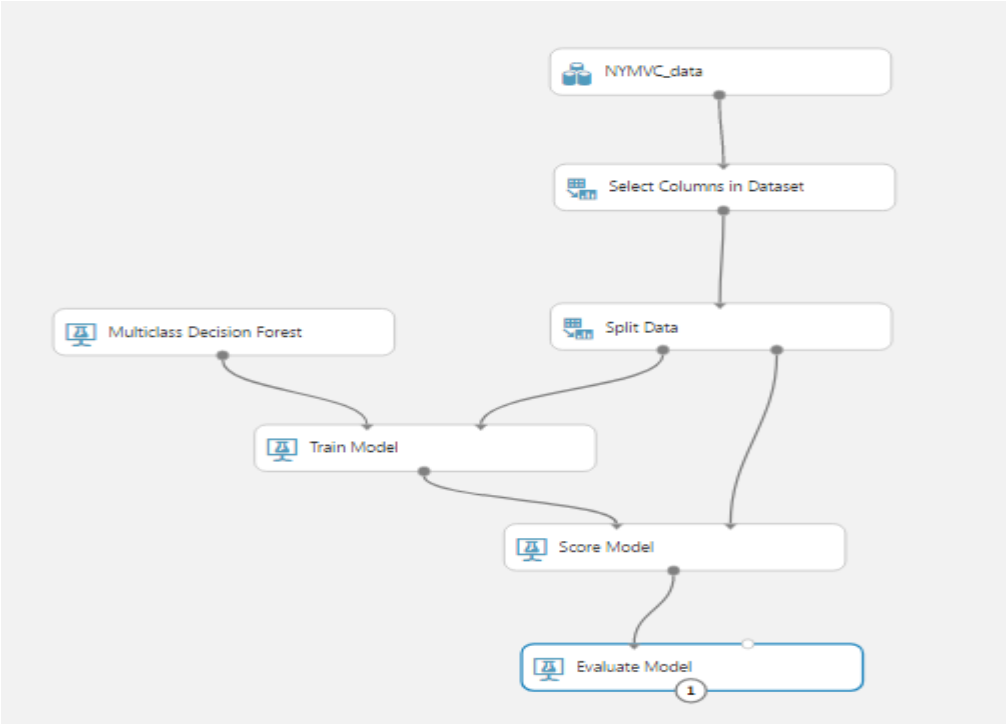


11. Search for the **Score Model** module and drag it onto the canvas.

12. Connect the **Trained Model** output port of the of the **Train Model** module to the **Trained Model** input port of the **Score Model** module. Connect the **Results dataset2** output port of the **Split Data** (Split) module to the **Dataset** port of the Score Model module.

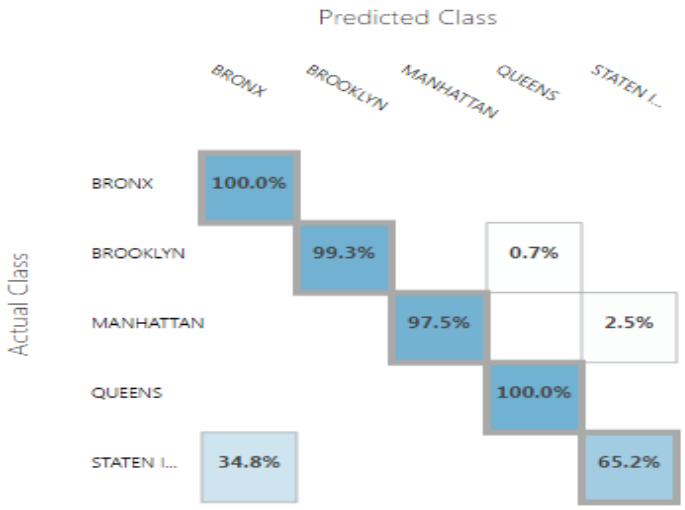


13. Search for the **Evaluate Model** module and drag it onto the canvas. Connect the **Scored Dataset** output port of the **Score Model** module to the left hand **Scored dataset** input port of the **Evaluate Model** module.



14. **Save** and **run** the experiment.

15. Click on **Evaluation Results** output port of **Evaluate Model** to visualize the **Multiclass Decision Forest** Module.



Metrics

Overall accuracy	0.975952
Average accuracy	0.990381
Micro-averaged precision	0.975952
Macro-averaged precision	0.945163
Micro-averaged recall	0.975952
Macro-averaged recall	0.924086

16. Search for the **Multiclass Neural Network** module. Drag this module onto the canvas. Set the Properties if this module as follows:
    a. **Create trainer mode**: Single Parameter
    b. **Hidden layer specification:** Fully connected case
    c. **Number of hidden nodes:** 100
    d. **The learning rate**: 0.1
    e. **Number of learning iterations**: 100
    f. **The initial learning weights diameter:** 0.1
    g. **The momentum:** 0
    h. **The type of normalizer:** Min-Max normalizer
    i. **Shuffle examples:** checked
    j. **Allow unknown categorical features:** checked

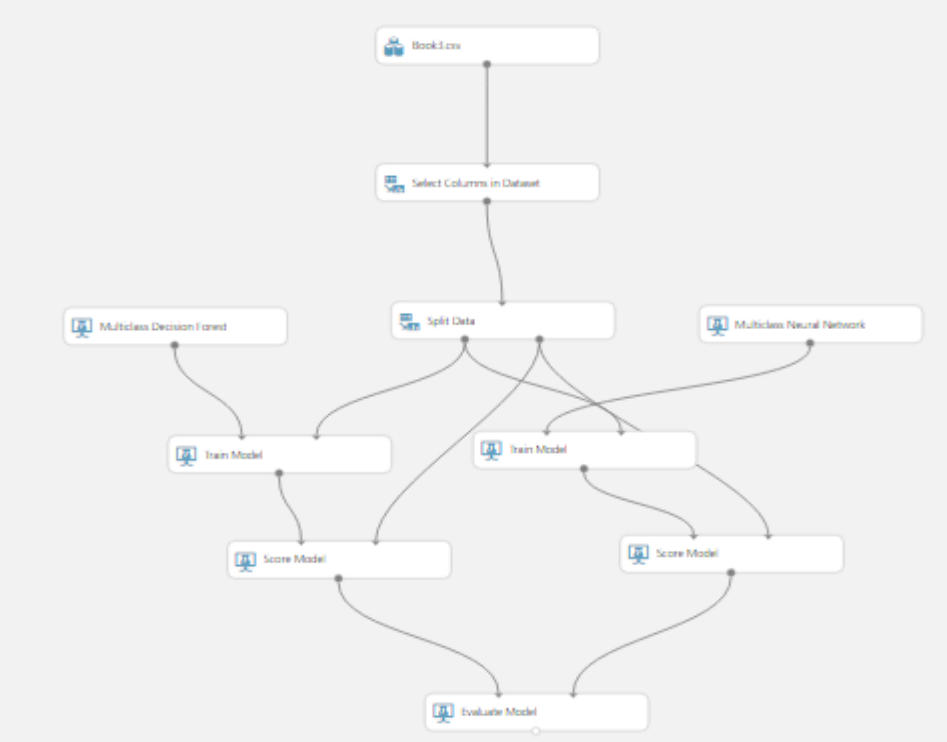
17. Search for the **Train Model** module. Drag this module onto the canvas.

18. Connect the **Untrained Model output** port of the **Linear Regression** module to the **Untrained Model** input port of the Train Model module. Connect the **Results dataset1** output port of the **Split Data** (Split) module to the **Dataset** input port of the Train model module. On the Properties pane, launch the column selector and select the **SCORE** column.

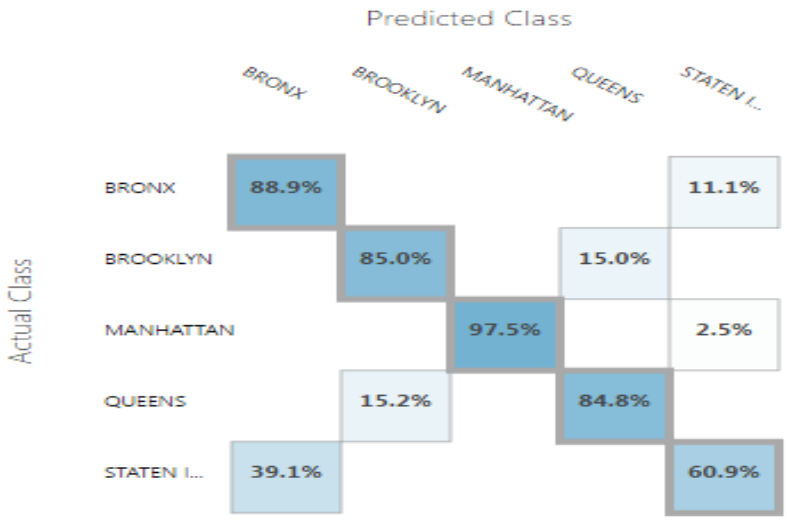
19. Search for the **Score Model** module and drag it onto the canvas.

20. Connect the **Trained Model** output port of the of the **Train Model** module to the **Trained Model** input port of the **Score Model** module. Connect the **Results dataset2** output port of the **Split Data** (Split) module to the **Dataset** port of the Score Model module.

21. Connect the **Scored dataset** output port of the **Score Model** module to the **Scored Dataset to Compare** input port of the **Evaluate Model**.



22. Click on **Evaluation Results** output port of **Evaluate Model** to visualize the **Multiclass Neural Network** Module.



Metrics

Overall accuracy	0.881764
Average accuracy	0.952705
Micro-averaged precision	0.881764
Macro-averaged precision	0.844263
Micro-averaged recall	0.881764
Macro-averaged recall	0.843747

23. Search for the **Multiclass Decision Jungle** module. Drag this module onto the canvas. Set the Properties if this module as follows
    a. **Resampling method: Bagging**
    b. **Create trainer mode:** Single Parameter
    c. **Number of decision DAGs**: 8
    d. **Maximum depth of the decision DAGs**: 32
    e. **Maximum width of the decision DAGs**: 128
    f. **Number of optimization:** 2048
    g. **Allow unknown values for categorical features:** Checked

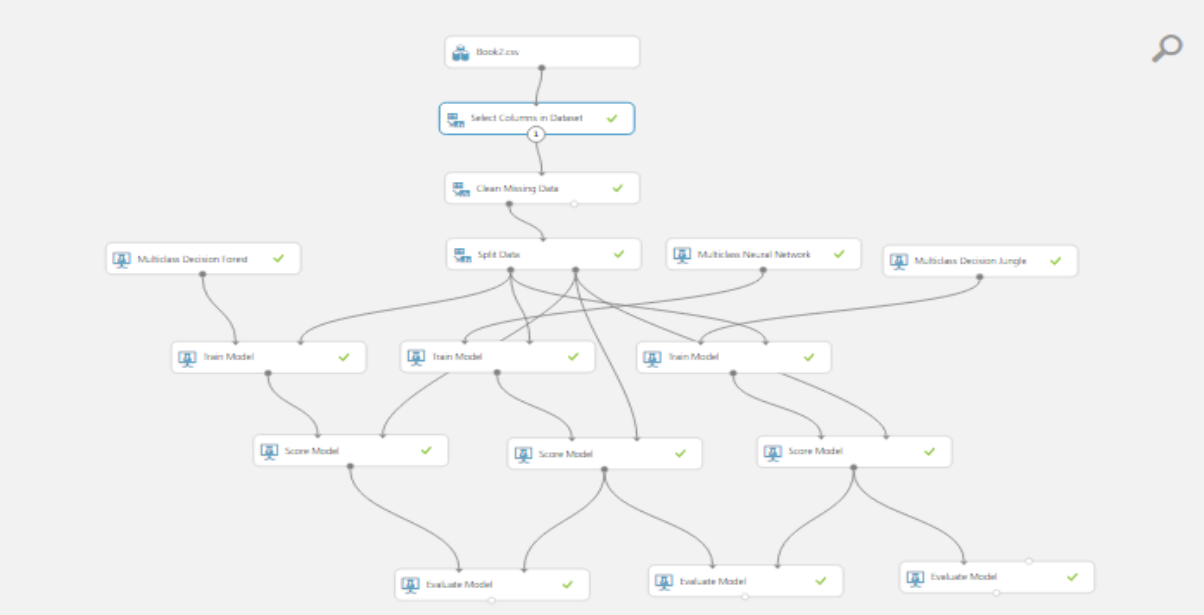
24. Search for the **Train Model** module. Drag this module onto the canvas.

25. Connect the **Untrained Model output** port of the **Linear Regression** module to the **Untrained Model** input port of the Train Model module. Connect the **Results dataset1** output port of the **Split Data** (Split) module to the **Dataset** input port of the Train model module. On the Properties pane, launch the column selector and select the **SCORE** column.

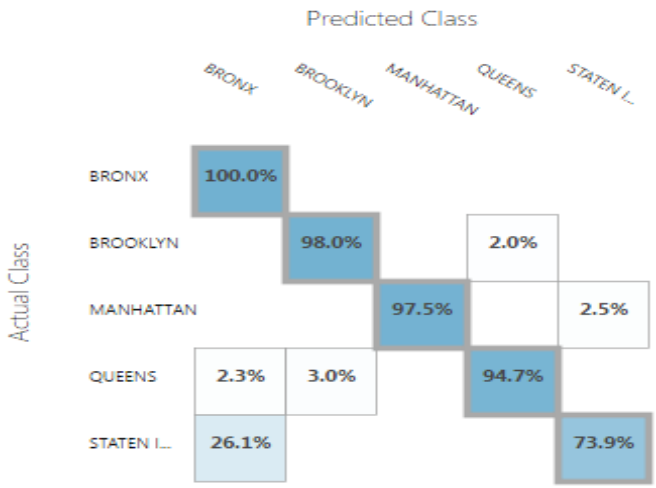
26. Search for the **Score Model** module and drag it onto the canvas.

27. Connect the **Trained Model** output port of the of the **Train Model** module to the **Trained Model** input port of the **Score Model** module. Connect the **Results dataset2** output port of the **Split Data** (Split) module to the **Dataset** port of the Score Model module.

28. Connect the **Scored dataset** output port of the **Score Model** module to the **Scored Dataset to Compare** input port of the **Evaluate Model**.



29. Click on **Evaluation Results** output port of **Evaluate Model** to visualize the **Multiclass Decision Jungle** Module. The **NYCMVC** experiment should look like the following.



Metrics

Overall accuracy	0.961924
Average accuracy	0.98477
Micro-averaged precision	0.961924
Macro-averaged precision	0.937895
Micro-averaged recall	0.961924
Macro-averaged recall	0.928256

Summary

In this lab, you have constructed and evaluated three multiclass classification model. Highlights from the results of this lab are:

MODEL	ACCURACY
Multiclass Decision Forest classification	97.59%
Multiclass Decision Jungle Classification	96.19%
Multiclass Neural Networks Classification	87.37%

Multiclass decision forest classification model turned out to be the best model out of the three as it had the best accuracy of 0.975952, precision of 0.945163 and recall of 0.924086.

References

1. https://studio.azureml.net/

2. Data set URL: https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95
3. GitHub link: https://github.com/anushamanjappa/CIS5560-Big-Data-Analysis