



CIS5560 Term Project Tutorial



Authors: Krithy Nanaiah Atrangada; Neha Shashidhara Guli; Anupam Sahay

Instructor: [Jongwook Woo](#)

Date: 05/16/2018

Lab Tutorial

katrang (katrang@calstatela.edu)

sneha (sneha@calstatela.edu)

asahay (asahay@calstatela.edu)

05/16/2018

San Francisco Bay Area Bike Share Analysis On Microsoft Azure Machine Learning

Objectives

List what your objectives are. In this hands-on lab, you will learn how to:

- Get data manually
- Create Spark cluster
- Train NLP system

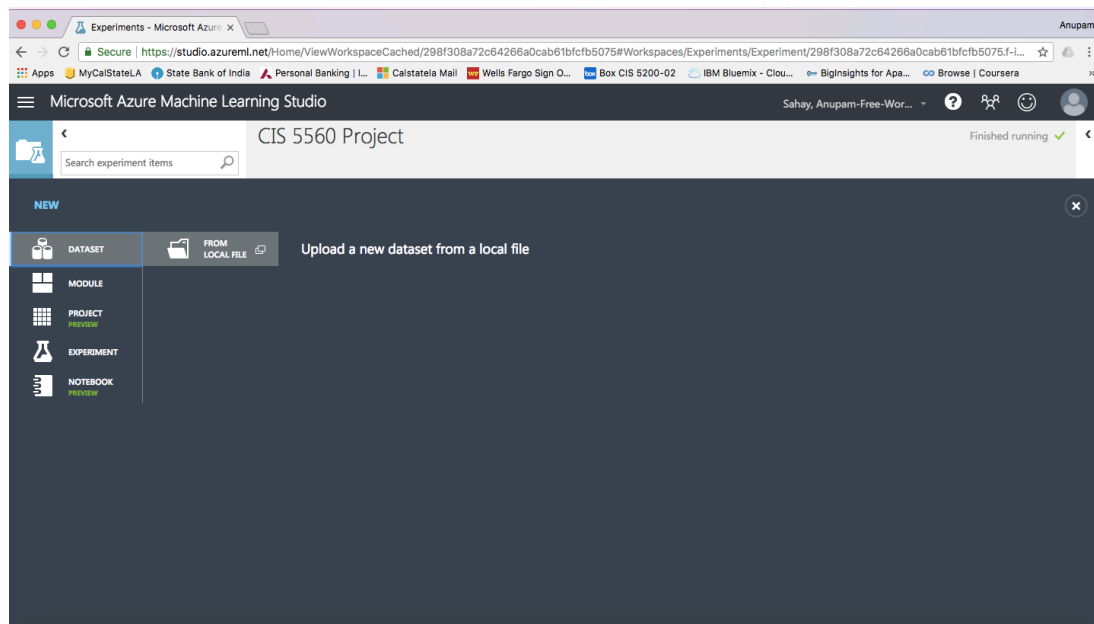
- Predicting total number of docks available in a station using Decision Forest Regression and Boosted Decision Tree.
- Visualization
- <https://gallery.cortanaintelligence.com/Experiment/CIS-5560-Project-2>

Platform Spec

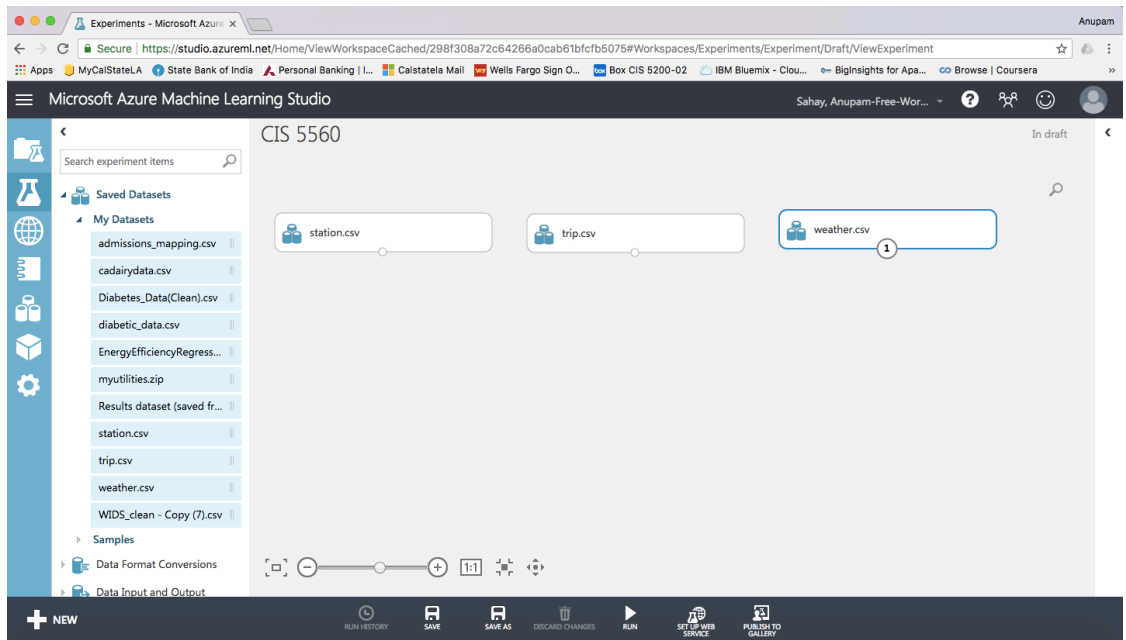
- Microsoft Azure Machine Learning
- CPU Speed: 3.4GHz
- # of nodes: 1
- Total Memory Size: 10GB

Step 1: Upload the Data Set from the Local File

1. This step is to upload the Dataset trip.csv, weather.csv, station.csv



- This dataset is available in the Kaggle website and was last updated 2 years ago
- We enter the name of the dataset which we have to work on i.e. trip.csv, weather.csv, station.csv in the in the working space.
- All the dataset should be in format of Generic CSV file with a header(.csv)



Step 2: Visualization of the Dataset Loaded in Azure ML

This step is to verify if all the columns are present in the dataset from source.

CIS 5560 > station.csv > dataset

id	name	lat	long	dock_count	city	installation_date
2	San Jose Diridon Caltrain Station	37.329732	-121.901782	27	San Jose	2013-08-06T00:00:00
3	San Jose Civic Center	37.330698	-121.888979	15	San Jose	2013-08-05T00:00:00
4	Santa Clara at Almaden	37.333988	-121.894902	11	San Jose	2013-08-06T00:00:00
5	Adobe on Almaden	37.331415	-121.8932	19	San Jose	2013-08-05T00:00:00
6	San Pedro Square	37.336721	-121.894074	15	San Jose	2013-08-07T00:00:00
7	Paseo de San Antonio	37.333798	-121.886943	15	San Jose	2013-08-07T00:00:00
8	San Salvador at 1st	37.330165	-121.885831	15	San Jose	2013-08-05T00:00:00

Microsoft Azure Machine Learning Studio

CIS 5560 > trip.csv > dataset

rows: 669959, columns: 11

id	duration	start_date	start_station_name	start_station_id	end_date	end_station_name
4576	63	2013-08-29T14:13:00	South Van Ness at Market	66	2013-08-29T14:14:00	South Van Ness at Market
4607	70	2013-08-29T14:42:00	San Jose City Hall	10	2013-08-29T14:43:00	San Jose City Hall
4130	71	2013-08-29T10:16:00	Mountain View City Hall	27	2013-08-29T10:17:00	Mountain View City Hall
4251	77	2013-08-29T11:29:00	San Jose City Hall	10	2013-08-29T11:30:00	San Jose City Hall
4299	83	2013-08-29T12:02:00	South Van Ness at Market	66	2013-08-29T12:04:00	Market
4927	103	2013-08-29T18:54:00	Golden Gate at Polk	59	2013-08-29T18:56:00	Golden Gate at Polk
4500	109	2013-08-29T13:25:00	Santa Clara at Almaden	4	2013-08-29T13:27:00	Adobe at Almaden
4563	111	2013-08-29T14:02:00	San Salvador at 1st	8	2013-08-29T14:04:00	San Salvador at 1st

Statistics

Visualizations

To view, select a column in the table.

Microsoft Azure Machine Learning Studio

CIS 5560 > weather.csv > dataset

rows: 3665, columns: 24

date	max_temperature_f	mean_temperature_f	min_temperature_f	max_dew_point_f
2013-08-29T00:00:00	74	68	61	61
2013-08-30T00:00:00	78	69	60	61
2013-08-31T00:00:00	71	64	57	57
2013-09-01T00:00:00	74	66	58	60
2013-09-02T00:00:00	75	69	62	61
2013-09-03T00:00:00	73	67	60	59
2013-09-04T00:00:00	74	68	61	59
2013-09-05T00:00:00	77	66	60	57

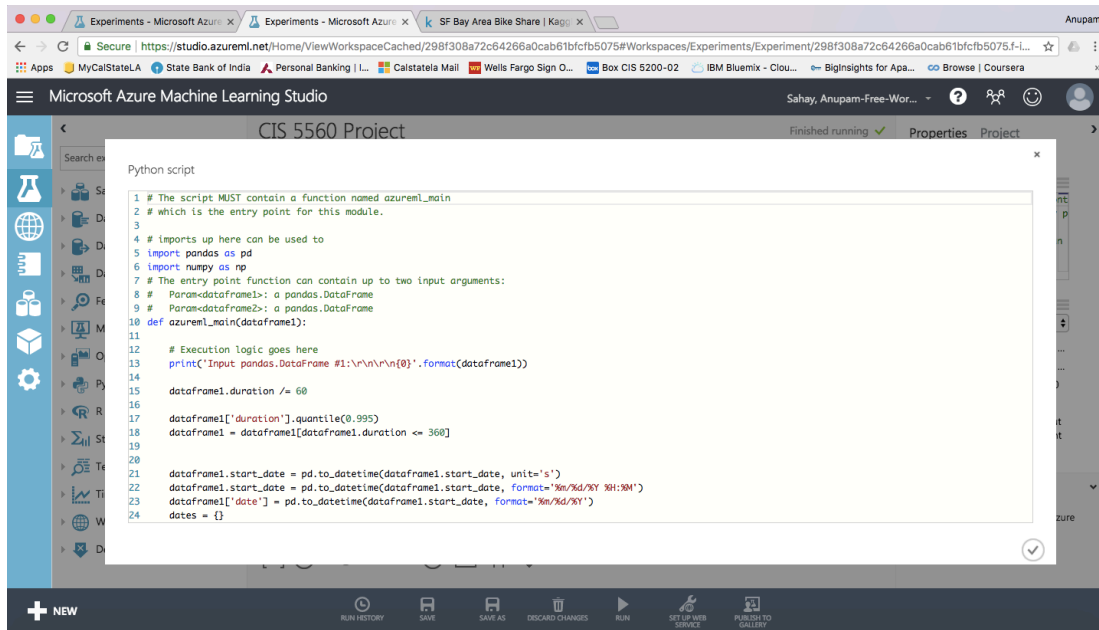
Statistics

Visualizations

To view, select a column in the table.

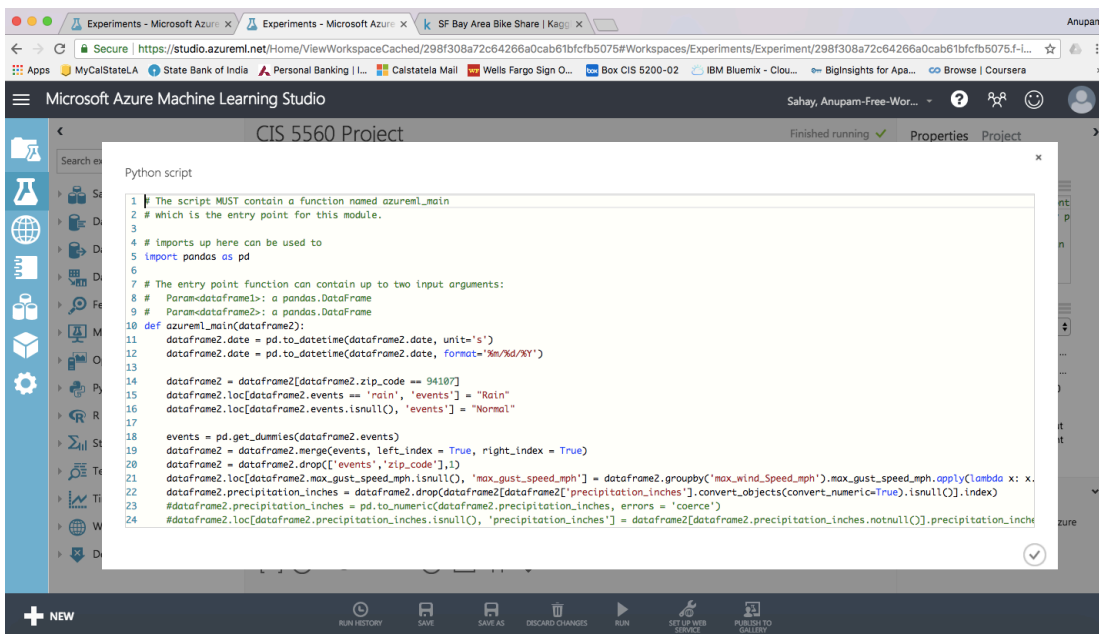
Step 3: Use Python Scripts

In this step we are using python scripts to drop the dummy values and to convert the duration into minutes and join all the tables.



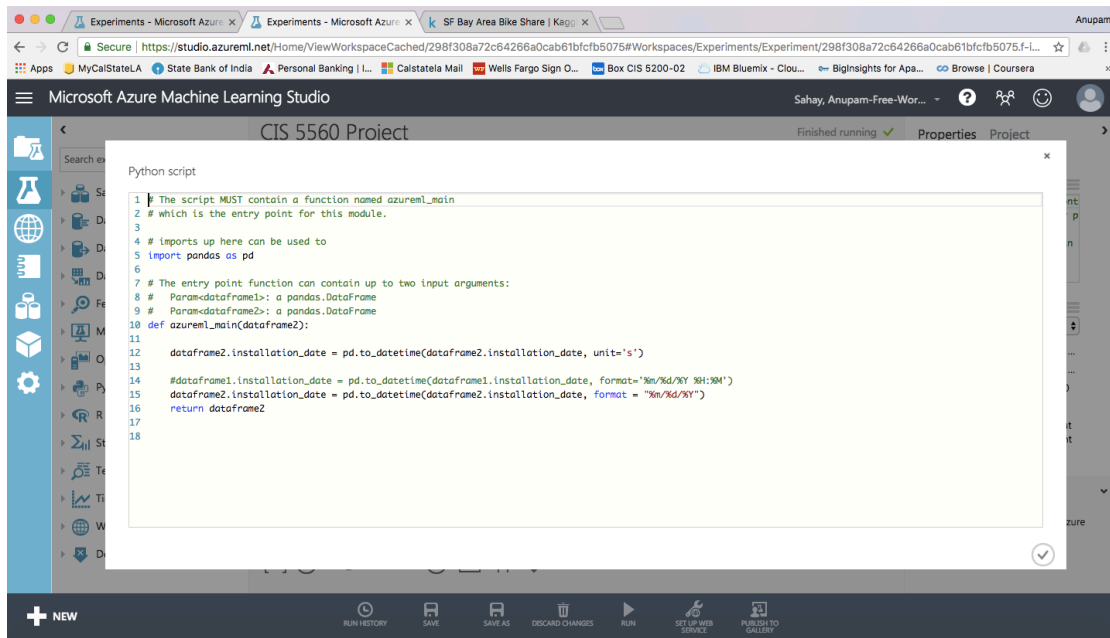
The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top bar indicates the user is 'Anupam' and the workspace is 'CIS 5560 Project'. The left sidebar contains various tool icons. The main area displays a Python script with the following code:

```
1 # The script MUST contain a function named azureml_main
2 # which is the entry point for this module.
3
4 # Imports up here can be used to
5 import pandas as pd
6 import numpy as np
7
8 # The entry point function can contain up to two input arguments:
9 # Param=DataFrame1: a pandas.DataFrame
10 # Param=DataFrame2: a pandas.DataFrame
11 def azureml_main(dataframe1):
12
13     # Execution logic goes here
14     print('Input pandas.DataFrame #1:\n\n{}\n'.format(dataframe1))
15
16     dataframe1.duration /= 60
17
18     dataframe1['duration'].quantile(0.995)
19     dataframe1 = dataframe1[dataframe1.duration <= 360]
20
21     dataframe1.start_date = pd.to_datetime(dataframe1.start_date, unit='s')
22     dataframe1.start_date = pd.to_datetime(dataframe1.start_date, format='%m/%d/%Y %H:%M')
23     dataframe1['date'] = pd.to_datetime(dataframe1.start_date, format='%m/%d/%Y')
24     dates = []
```



The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top bar indicates the user is 'Anupam' and the workspace is 'CIS 5560 Project'. The left sidebar contains various tool icons. The main area displays a Python script with the following code:

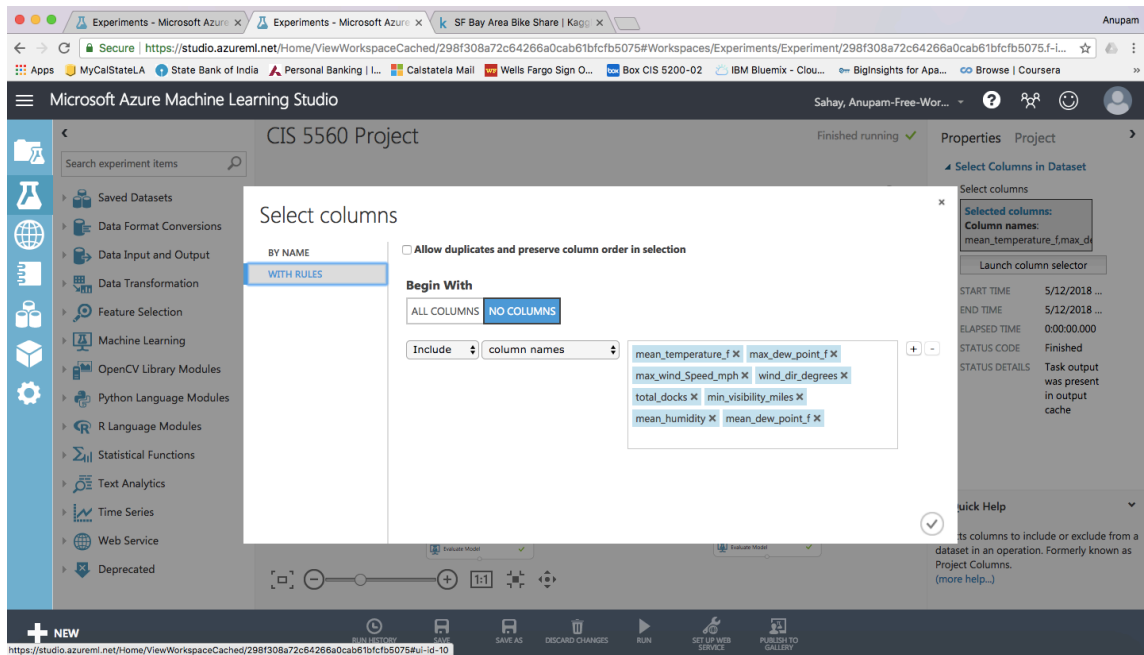
```
1 # The script MUST contain a function named azureml_main
2 # which is the entry point for this module.
3
4 # Imports up here can be used to
5 import pandas as pd
6
7 # The entry point function can contain up to two input arguments:
8 # Param=DataFrame1: a pandas.DataFrame
9 # Param=DataFrame2: a pandas.DataFrame
10 def azureml_main(dataframe2):
11     dataframe2.date = pd.to_datetime(dataframe2.date, unit='s')
12     dataframe2.date = pd.to_datetime(dataframe2.date, format='%m/%d/%Y')
13
14     dataframe2[dataframe2.zip_code == 94107]
15     dataframe2.loc[dataframe2.events == 'rain', 'events'] = "Rain"
16     dataframe2.loc[dataframe2.events.isnull(), 'events'] = "Normal"
17
18     events = pd.get_dummies(dataframe2.events)
19     dataframe2 = dataframe2.merge(events, left_index = True, right_index = True)
20     dataframe2 = dataframe2.drop(['events', 'zip_code'], 1)
21     dataframe2.loc[dataframe2.max_gust_speed_mph.isnull(), 'max_gust_speed_mph'] = dataframe2.groupby('max_wind_speed_mph').max_gust_speed_mph.apply(lambda x: x)
22     dataframe2.precipitation_inches = dataframe2.drop(dataframe2[dataframe2['precipitation_inches'].convert_objects(convert_numeric=True).isnull()].index)
23     #dataframe2.precipitation_inches = pd.to_numeric(dataframe2.precipitation_inches, errors = 'coerce')
24     #dataframe2.loc[dataframe2.precipitation_inches.isnull(), 'precipitation_inches'] = dataframe2[dataframe2.precipitation_inches.notnull()].precipitation_inches
```



- All the 3 tables are joined using the python scripting
- The dummies and the redundant data are dropped
- All the data which has no effect on our prediction column are removed and the size of the dataset is reduced.
- We are using only 57 rows and 8 columns.

Step 4: Select columns in Data Set

This is a common interface in Azure Machine Learning modules to enable selecting the columns you want to use in the experiment, in our case column total_docks. In the select column dialog box, select option with Rule to begin with no columns, and include all the column name shown in the image.



Properties:

- We are selecting the column in the Dataset using Permutation feature Importance which is affecting our prediction column.
- We then do two models to predict total_docks to find out the best prediction model we have developed.

Step 5: Model 1

- In this model we have used the Boosted Decision Tree Algorithm

Create Trainer Mode: ingle Parameter

Maximum Number of leaves: 20

Minimum Number of samples: 10

Learning Rate: 0.006

Total Number of trees Connected: 1000

Random Number of seeds: 0

Allow Unknown Category: Checked

- We split the data in the in the following order: -

Splitting Mode: Split rows

Fraction of rows in the first output dataset: - 0.7

Randomized Split: Checked

Random Seed: 0

Stratified Split: False

- Cross-Validation Model: Selected Column, With Rules Include column names: "total_docks",

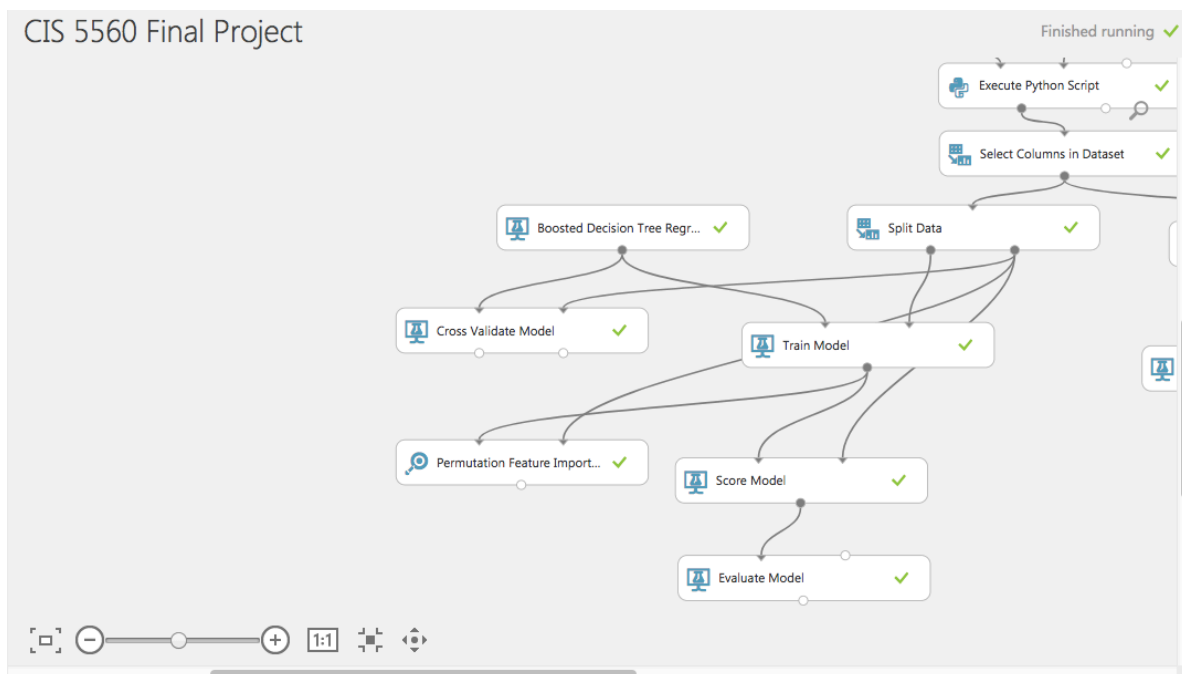
Random Seed: Checked

- In Train Model: Selected Column, With Rules Include column names: "total_docks"

- In Permutation Feature Importance:

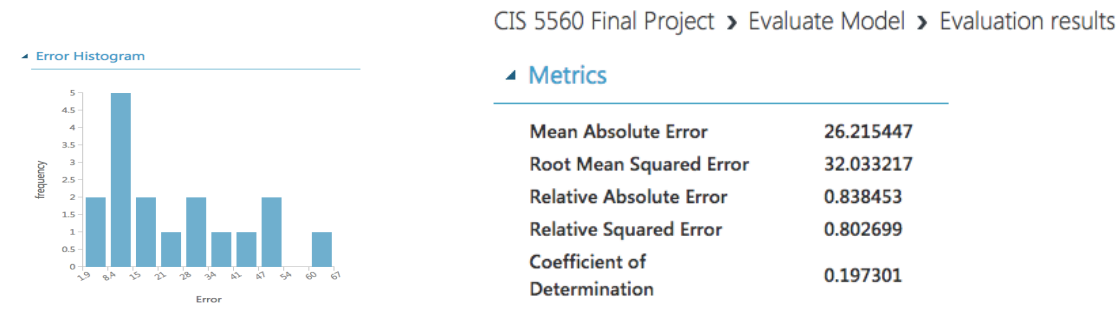
Random Seed: 0

Metric for measuring performance: Regression: Root mean square error



Step 6: Visualization-1

These are the results which we achieved for the first model:



Step 7: Model 2

- In this model we have used the Decision Forest Regression Algorithm

Resampling Method: Bagging

Create Trainer Mode: Single Parameter

Number of decision Trees: 8

Maximum Depth of the decision forest: 32

Number of Random Splits: 128

Minimum number of samples: 1

Allow Unknown Values: Checked

- We split the data in the in the following order: -

Splitting Mode: Split rows

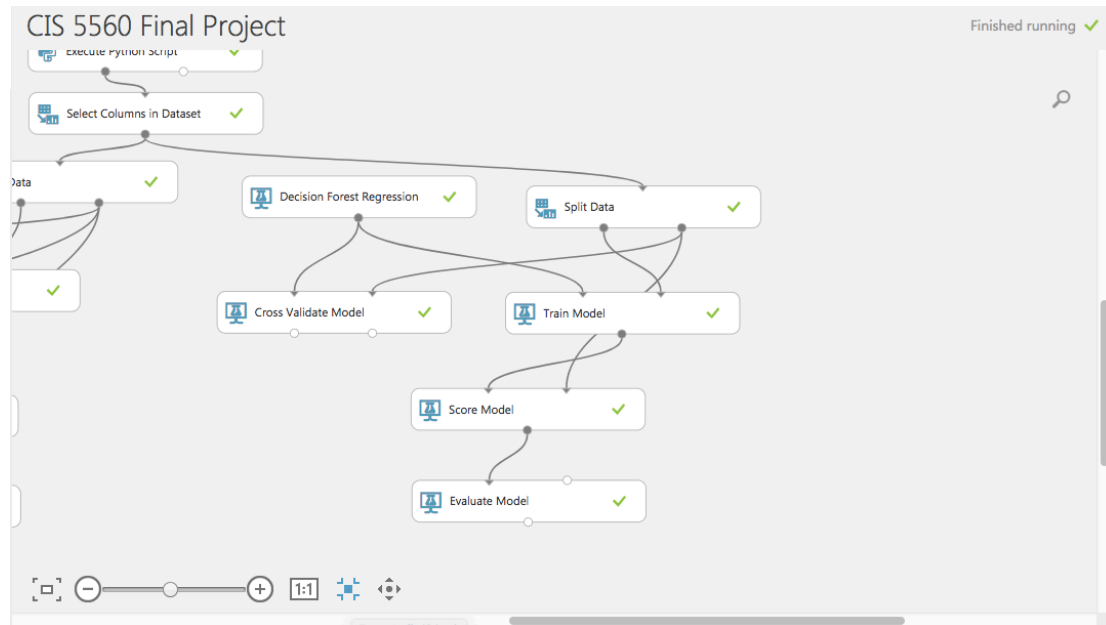
Fraction of rows in the first output dataset: - 0.7

Randomized Split: Checked

Random Seed: 0

Stratified Split: False

- Cross-Validation Model: Selected Column, With Rules Include column names: “total_docks”,
Random Seed: Checked
- In Train Model: Selected Column, With Rules Include column names: “total_docks”





Step 7: Visualization-2

These are the results which we achieved for the second model:

CIS 5560 Final Project > Evaluate Model > Evaluation results

rows
1

columns
6

view as
 

	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
	73.657657	20.063725	29.892925	0.641702	0.699018	0.300982

>

Statistics

Mean	0.301
Median	0.301
Min	0.301
Max	0.301
Standard Deviation	NaN
Unique Values	1
Missing Values	0
Feature Type	Numeric Feature


Visualizations

Coefficient of Determination

Histogram

compare to

None



References:-

1. <https://www.kaggle.com/benhamner/sf-bay-area-bike-share/data> - 4.45GB
2. <https://gallery.cortanaintelligence.com/Experiment/CIS-5560-Project-2>