**Problem Statement and Use Case**

The project aims to predict and classify if for a given call received by the EMS they would not be able to assign the EMS unit immediately. Based on the summary metrics of the data 8% of the calls received by the EMS are not assigned with the EMS unit. Critical cases have been addressed late which could be life threatening.

This will act as a performance measure for the EMS to improve the service response to cases received. Once this is available from the historical data, we can determine the resource allocation on a given day, time and the location in the city. This will help to reduce the response time as well as attend the urgent calls such as cardiac arrests or any fatal health conditions.

**Dataset**

The data set talks about the call logs received by the EMS service of the city of New York which acts with the Fire Department of New York City (FDNY) combined. The data contains information about the Call types, severity, location of the incident and the time responses of the EMS service to address the issue. The dataset is available on the NYC Open Data website :

[Dataset URL](https://data.cityofnewyork.us/Public-Safety/EMS-Incident-Dispatch-Data/76xm-jjuj)

Rows : 4827514

Cols : 32

**Data pre-processing**

- The first process of cleaning the data involved removal of the incident id column which is just the unique id of the row.

Columns dropped : CAD\_INCIDENT\_ID

- Next were the columns which were post the Unit assignment, our target being the cases where the units were not assigned to the incident.

Columns dropped :

'FIRST\_HOSP\_ARRIVAL\_DATETIME','FIRST\_TO\_HOSP\_DATETIME'

- There were columns which provided us the Geographical aspects of the NY city for different entities eventually kept only the column which helped to classify the HELD\_INDICATOR.

Columns dropped : 'POLICEPRECINCT','CITYCOUNCILDISTRICT','COMMUNITYDISTRICT','COMMUNITYSCHOOLDISTRICT','CONGRESSIONALDISTRICT'

- Other Columns dropped involved timestamps of intermediate activities

Columns dropped : 'INCIDENT\_DATETIME','FIRST\_ASSIGNMENT\_DATETIME','FIRST\_ACTIVATION\_DATETIME','FIRST\_ON\_SCENE\_DATETIME','INCIDENT\_CLOSE\_DATETIME'

NGRESSIONALDISTRICT'

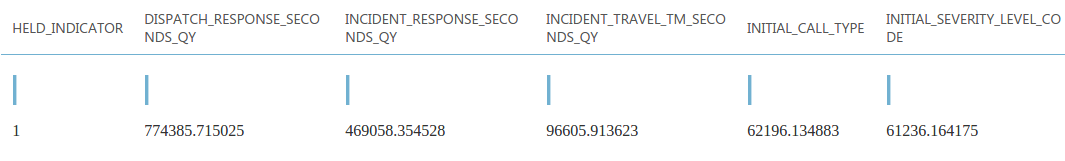
- Dropped columns which were nulls for large amount of rows

Columns dropped : 'ATOM'

- Derived a column : CALL\_SEVERITY\_TYPE, based on the Call severity level to reduce the number of factors.

**Feature Selection :**

- Performed the Chi Squared Analysis for feature selection



- This method gave us selected columns to be used for the data modelling.

**Luigi**

**Data flow**

- Luigi framework will be used to incorporate all the below task

Task1 - Download the dataset from the URL

Task2 - Data Pre-processing

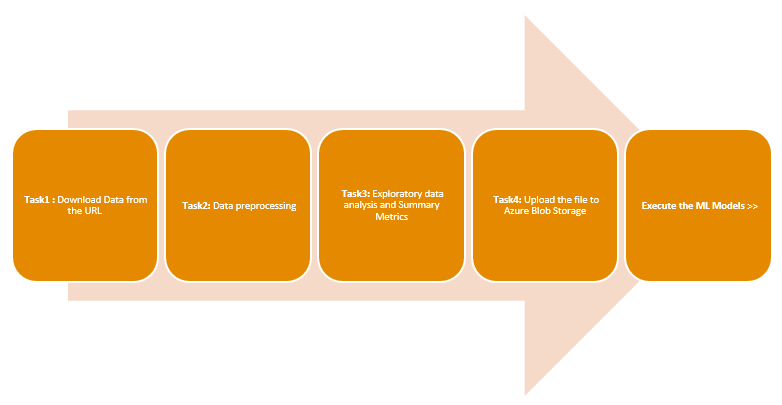
Task3 - Exploratory data analysis and generate summary metrics

Task4 - Upload the generated file to the MS Azure Storage.

**Docker and EC2 Deployment**

- The luigi script will be available on the docker Image (*bhavikv9488/adsteam04*) and this will be deployed on the AWS ec2 instance.

- Once the ec2 instance is started, the image can be executed automatically and based on the cron schedule within the docker image the script will be executed to upload the output file to Azure.



- This Luigi Pipeline has been schedule on the docker image which will run at the end of each day

- Once this data is available on the Microsoft Blog Storage, it can be available to the Azure ML studio by importing it to the models/experiment.

**Clustering**

- Before we deploy the models, we have divided the data into clusters which will helps us provide results with better prediction and accuracy. The clustering methods implemented are :

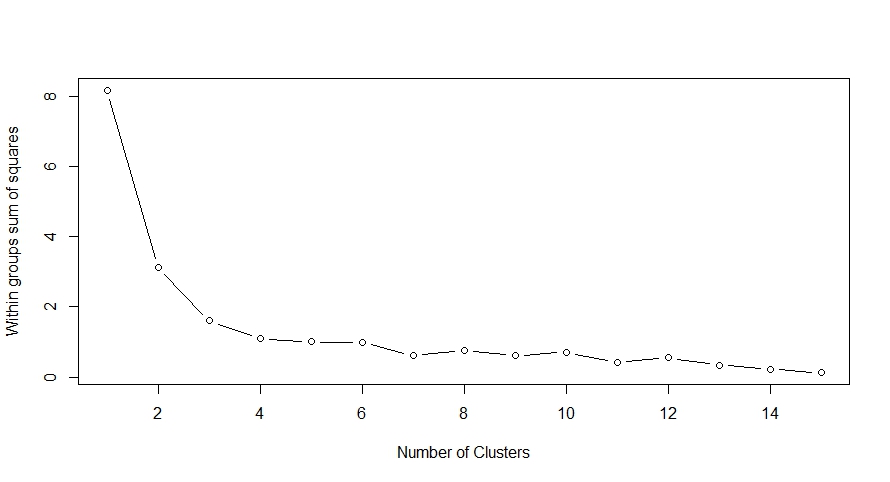
**1. Manual Clustering** : Created 3 Clusters based on the call severity categories as follows

- **Category High** having calls from severity level 1 to 3.

- **Category Medium** having calls from severity level 4 to 6.

- **Category Low** having calls from severity level 7 to 10.

1. **Clustering using K-means** : Created 3 clusters based on the K-Means algorithm



1. **Entire Dataset as a cluster** : Execute the ML models for the entire dataset as a cluster.



**Azure ML Studio**

- Once the data is available on the blob storage, it can be imported to the ML studio experiments and ML models can be deployed to be executed on the given data. This data will be divided into train and test. Models will be trained on the train data and test data can be available as a score.

- The algorithms which were implemented for the classification purpose are

- **Logistic Regression**

**- Decision Tree**

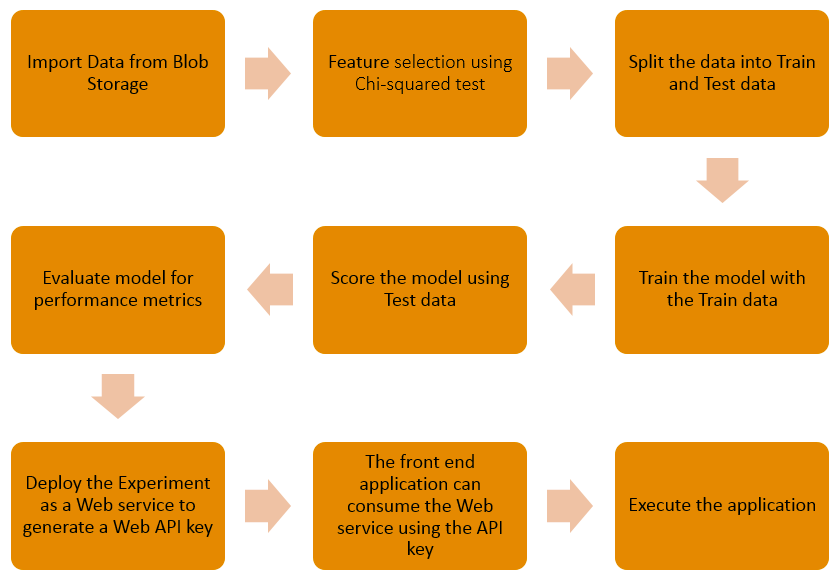
**- Neural Network**

**- Support Vector Machines**

**Process Flow of the Application**

- The below flow will be excuted for all the clusters defined on the Azure studio whenever a request has been received from the application.

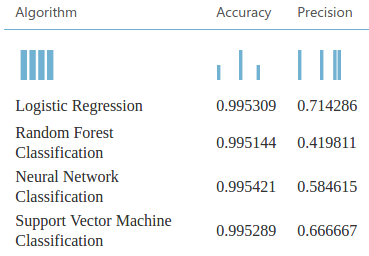
- The Model will return the flag values as Yes or No as a classified value



**Model performance metrics : Method 1 : Manual Clustering**

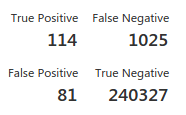
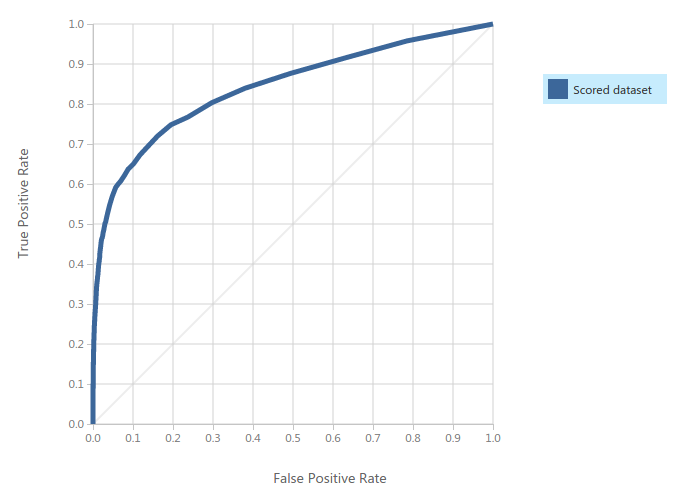
**Cluster : High Severity Level**

Measure for all algorithms :



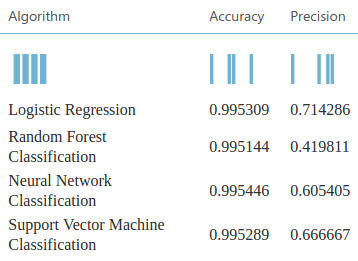
- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **Neural Network.**

**ROC Curve and the Confusion matrix for Neural network for High Severity cluster**



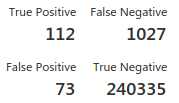
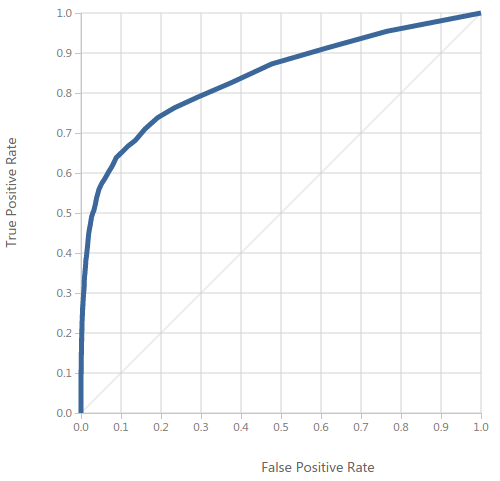
**Cluster : Medium Severity Level**

**Measure for all algorithms :**



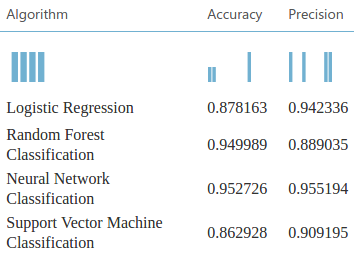
- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **Neural Network.**

**ROC Curve and the Confusion matrix for Neural network for Medium Severity Cluster**



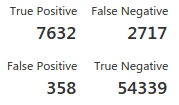
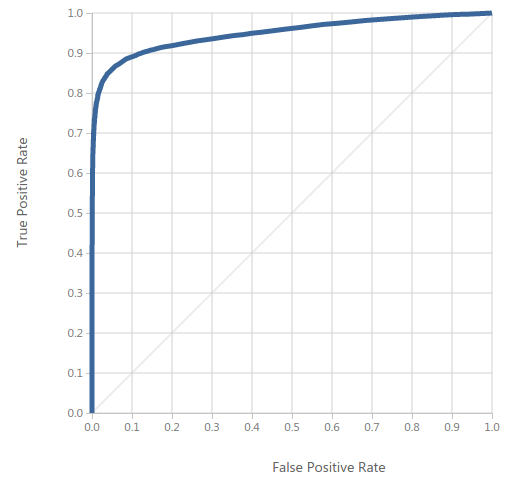
**Cluster : Low Severity Level**

**Measure for all algorithms :**



- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **Neural Network.**

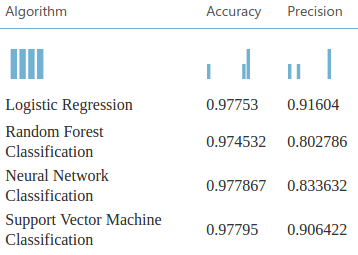
**ROC Curve and the Confusion matrix for Neural network for Low Severity Cluster**



**Model performance metrics : Method 2 : K-Means Clustering ( 3 Clusters)**

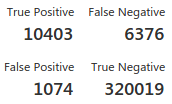
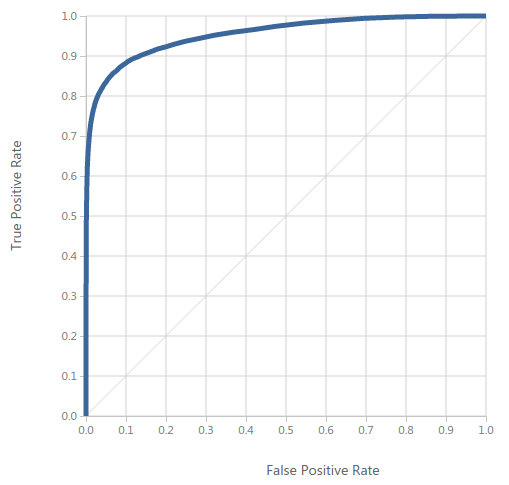
**Cluster 1:**

**Measure for all algorithms :**



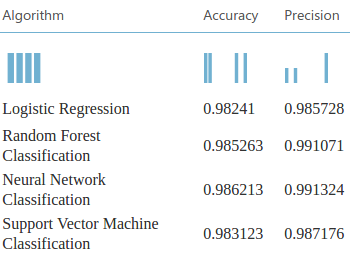
- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **SVM.**

**ROC Curve and the Confusion matrix for SVM for Cluster 1 :**



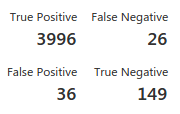
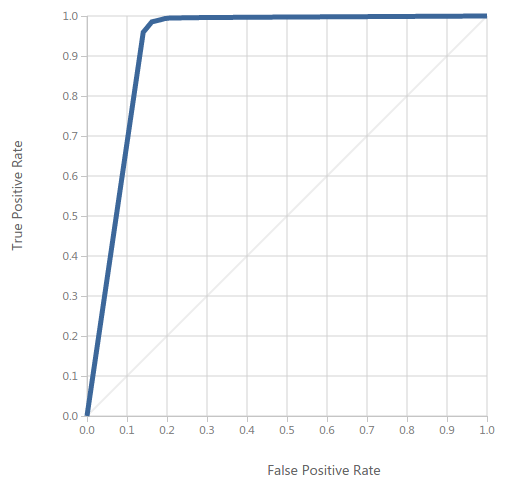
**Cluster 2:**

**Measure for all algorithms :**



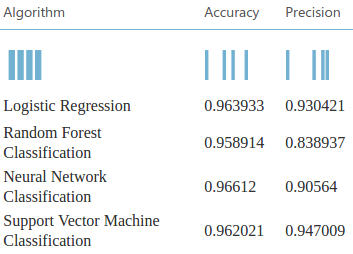
- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure and confusion metrics for this cluster is the **Random Forest Algorithm.**

**ROC Curve and the Confusion matrix for Random Forest Algorithm for Cluster 2 :**



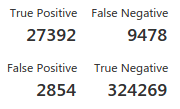
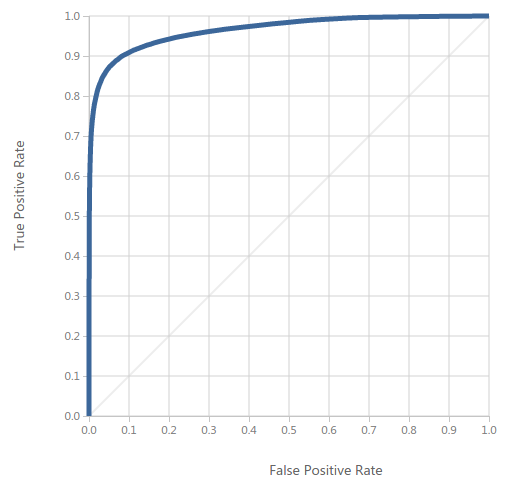
**Cluster 3:**

**Measure for all algorithms :**



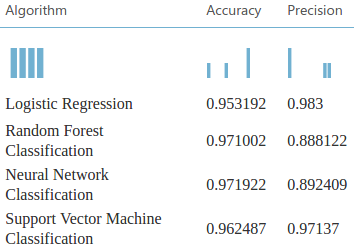
- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **Neural Network Algorithm.**

**ROC Curve and the Confusion matrix for Neural network for Cluster 3 :**



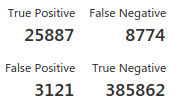
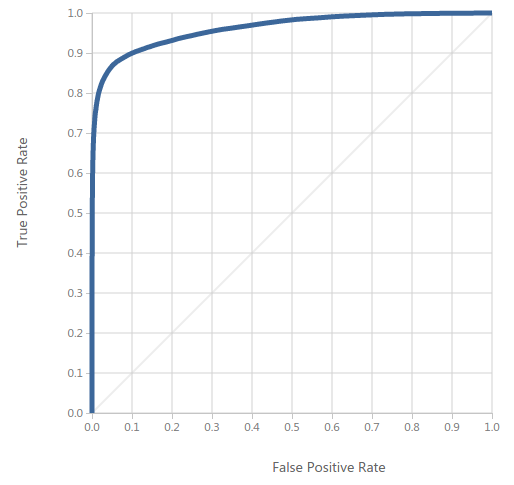
**Model performance metrics : Method 3 : Entire Dataset as a cluster :**

**Measure for all algorithms :**



- After executing all the 4 algorithms on this cluster the above results can be interpreted and the best suitable algorithm based on the accuracy measure for this cluster is the **Neural Network Algorithm.**

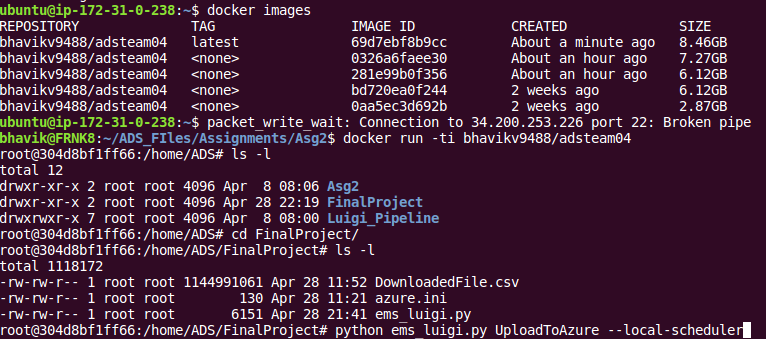
**ROC Curve and the Confusion matrix for Neural network for Entire Cluster :**



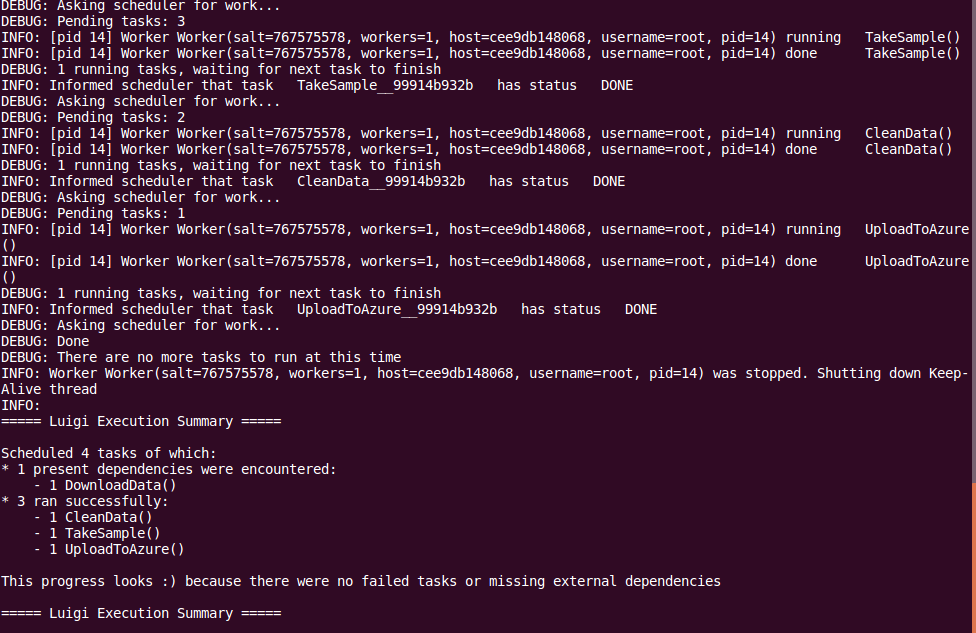
**Execution Steps for the application**

- Once the ec2 instance is started and logged on, run the docker image (bhavikv9488/adsteam04).

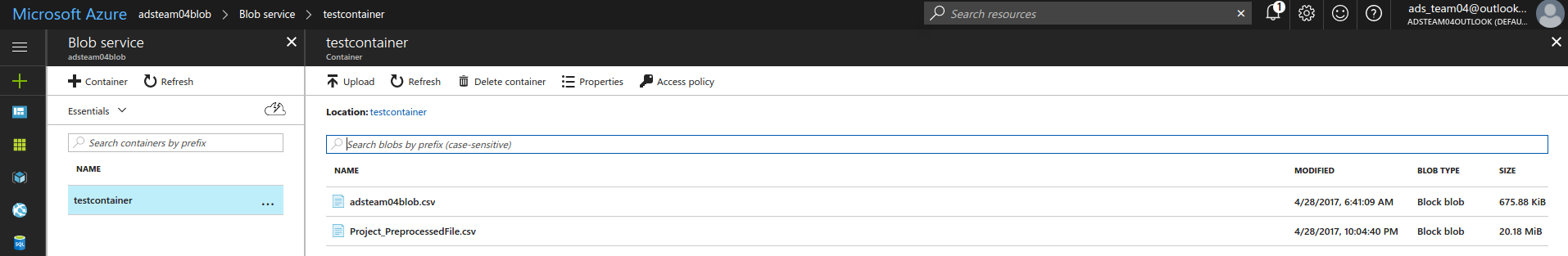
- The Luigi script files are available in the /home/ADS/FinalProject directory on the docker image.



- Execute the Luigi script as below and the progress can be checked on the console.



- The files will now be available on the Azure Blob Storage.



To log on to the ML studio and view the experiments and the Blob files using the shared credentials.