Final Report

# Flights delay prediction

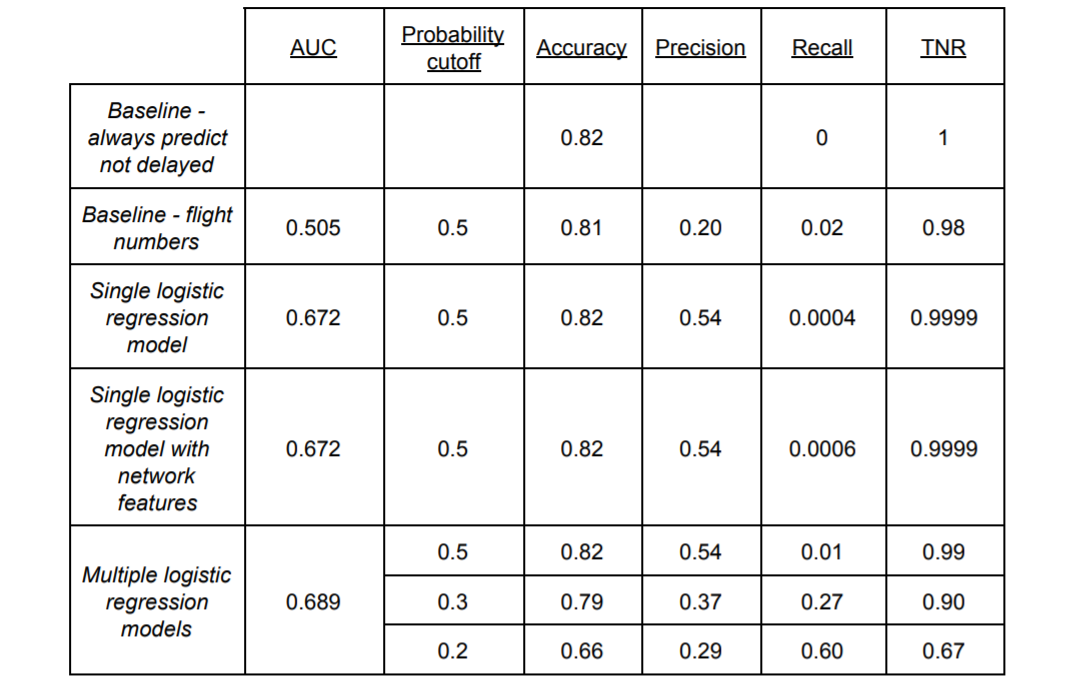
## Pruthvij Thakar

## Sumedh Saraf

## (Team 4)

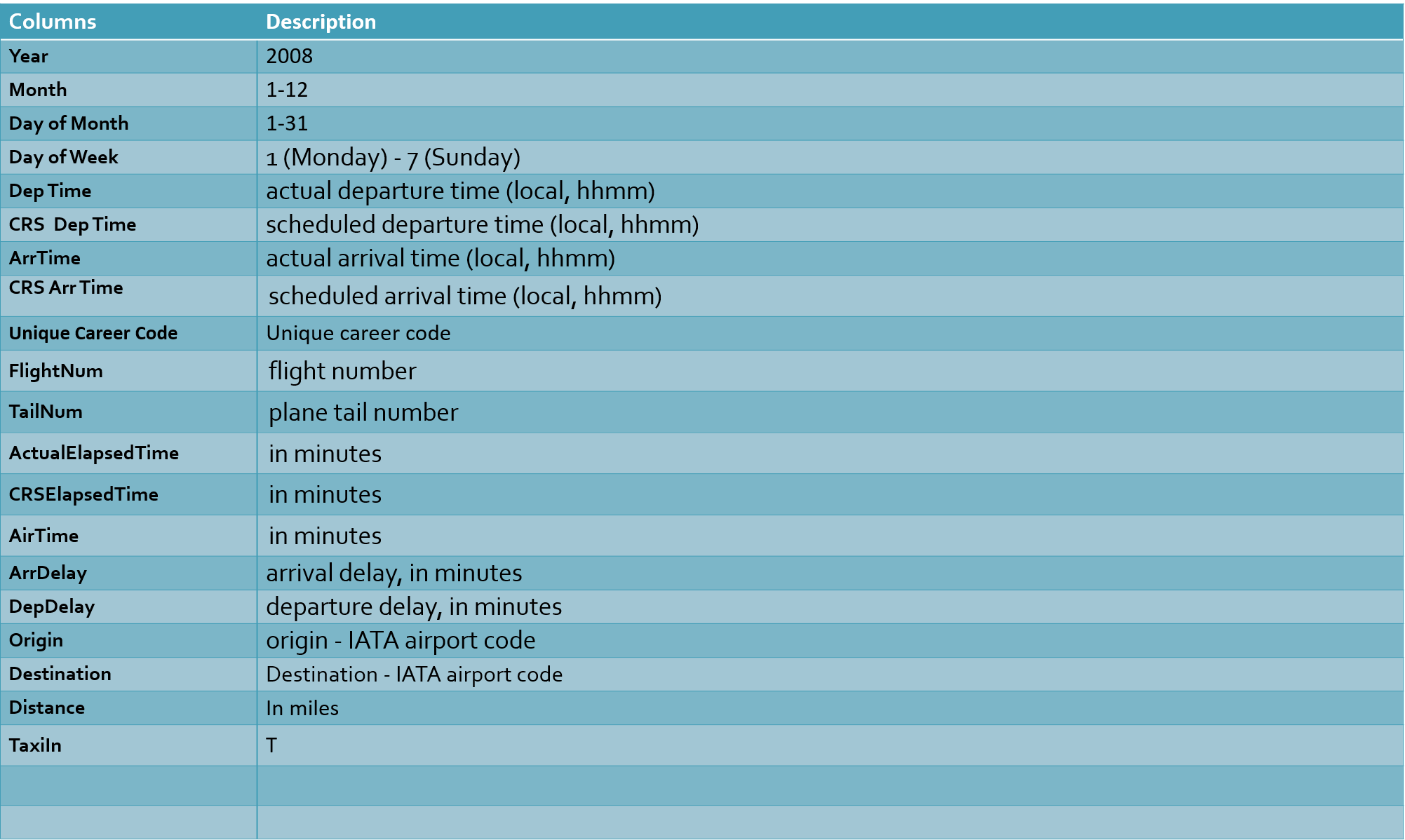
### Problem Statement:

The flights dataset we have taken is very challenging in terms of obtaining a good accuracy and classifying the non-delayed flights correctly with both the cases prediction and classification. It remains a huge challenge for the industry as below you can see a report of best results that people have derived in past trying out different approaches based on different algorithms and different feature engineering.



Reference: <https://srcole.github.io/assets/flight_delay/report.pdf>

1. Data Source: 1) <http://stat-computing.org/dataexpo/2009/the-data.html>



2) Historical weather and flight demand data for 2008 is from the FAA Aviation Systems Performance Metrics (ASPM) (<https://aspm.faa.gov/>)

1. Data Pipelines:

**Part 1**

Perform best algorithm

Docker image

Upload to AWS S3

Data ingestion

Confusion Matrix

EDA

**Part 2**

Feature Engineering and

Variable Selection

Apply Classification

Algorithms

Data Ingestion

Data Cleaning

Deployed on Azure Portal

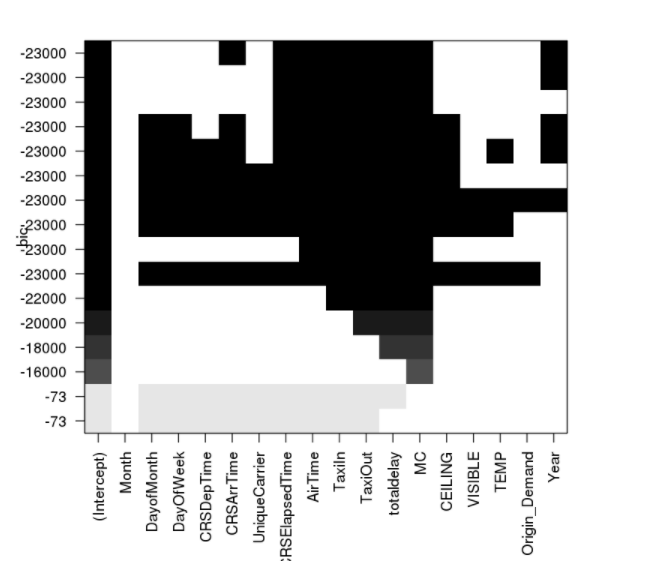
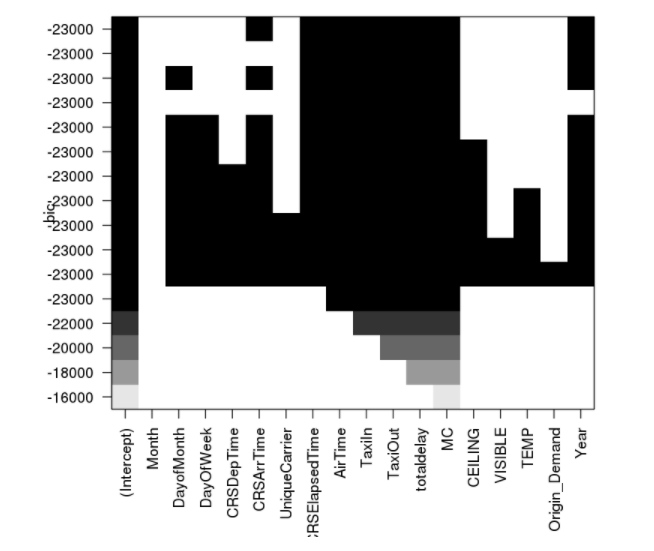
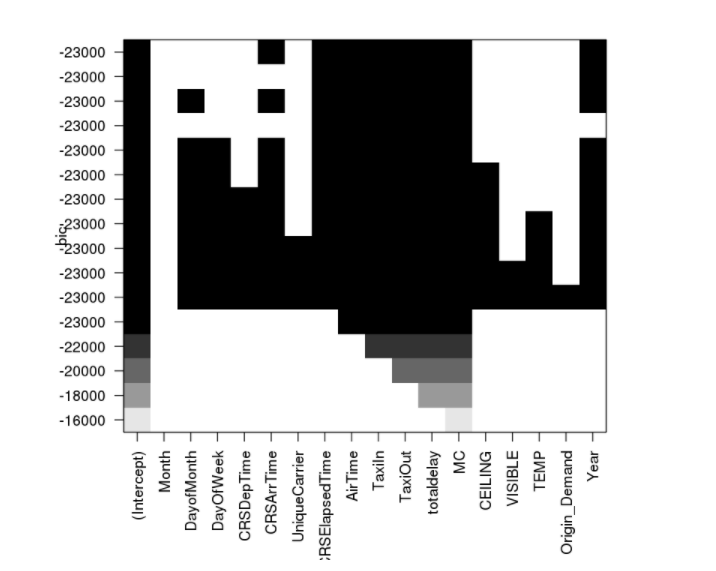
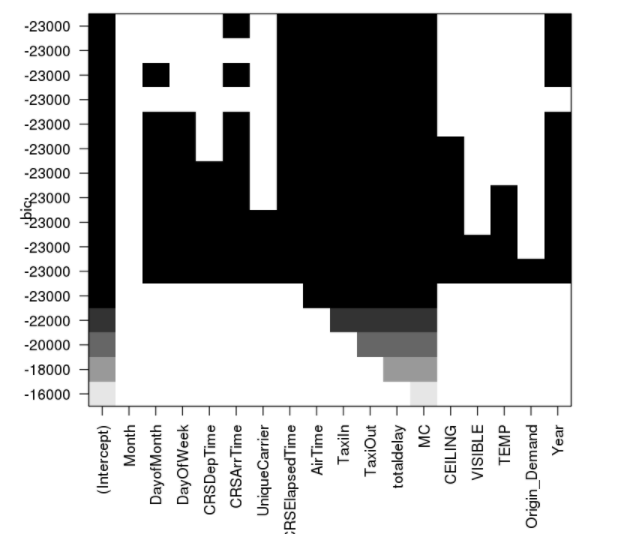
Build algorithm on Azure ML with python scripts

Apply Regression Algorithms

1. Data Cleaning: The data included 29 columns and more than 8 million records for the flights of year 2008 from Bureau of Transportation statistics. We have also considered the Weather data of the same year where in you can see the weather of the area where in the weather of that area is given.
2. Feature Engineering and Variable Selection: Feature engineering has shown to be very prominent part of the project as we have derived many columns from the given columns like the total delays is derived from various delays mentioned in the various columns.

Used R for variable selection :

The following images represents the results of the various search methods commonly used to compute the variable selection i.e. Exhaustive, SerRep, Forward and Backward search methods respectively.



Pearson co efficient

 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation

DepDelay totaldelay MC VISIBLE TEMP Month Origin\_Demand CRSDepTime UniqueCarrier CRSArrTime AirTime DayofMonth TaxiOut TaxiIn CEILING CRSElapsedTime DayOfWeek

1 0.974021 0.183692 0.177052 0.127921 0.106165 0.097805 0.081553 0.080685 0.078766 0.062505 0.053468 0.05134 0.039574 0.034005 0.022009 0.003253

Spearmen coefficient

It assesses how well the relationship between two variables can be described using a monotonic function.

The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables; while Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not). If there are no repeated data values, a perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other.

A Spearman correlation of 1 results when the two variables being compared are monotonically related, even if their relationship is not linear. This means that all data-points with greater x-values than that of a given data-point will have greater y-values as well. In contrast, this does not give a perfect Pearson correlation.

DepDelay totaldelay MC CEILING VISIBLE UniqueCarrier CRSDepTime TEMP CRSArrTime Month AirTime Origin\_Demand DayofMonth TaxiOut

CRSElapsedTime DayOfWeek TaxiIn

1 0.886809 0.179367 0.168102 0.141257 0.119386 0.114911 0.111294 0.111143 0.089669 0.069702 0.051029 0.049072 0.045426 0.026604 0.006572 0.004164

Statistics

Fisher Coefficient

The Fisher information (sometimes simply called information[1]) is a way of measuring the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X. Formally, it is the variance of the score, or the expected value of the observed information.

DepDelay totaldelay VISIBLE MC TaxiIn AirTime Origin\_Demand Month CRSDepTime CRSArrTime UniqueCarrier DayofMonth TaxiOut DayOfWeek TEMP CRSElapsedTime CEILING

1 20.951518 0.15652 0.112053 0.10566 0.08253 0.066527 0.066502 0.065912 0.064858 0.059307 0.052098 0.051914 0.050981 0.050529 0.048817 0.037955

Chi squared tests

DepDelay totaldelay CRSArrTime CRSDepTime CEILING Month MC VISIBLE Origin\_Demand TEMP UniqueCarrier AirTime DayofMonth TaxiOut DayOfWeek CRSElapsedTime TaxiIn

view as

1 11667.978148 384.309731 369.913835 358.038812 320.412586 224.320598 197.215887 182.739383 176.705769 167.706589 153.349124 135.259039 128.818648 103.731047 87.673697 77.670935

# Introducing the weather data and then calculating the coefficient values of different features

Pearson Correlation

DepDelay totaldelay T\_O\_GATE\_DELAY\_A T\_GATE\_DELAY\_A OAG\_ARPT\_DEP\_15\_A T\_OAG\_ARPT\_DEP\_A T\_O\_GATE\_DELAY O\_GATE\_DEL\_15\_A T\_PTM\_ARPT\_DEP\_A T\_GATE\_DELAY OAG\_ARPT\_DEP\_15M T\_OAG\_ARPT\_DEP O\_GATE\_DEL\_15\_PCNT DEL\_DEP15\_PCNT T\_ARR\_EDCT\_HOLD O\_GATE\_DEL\_15C O\_GATE\_DEL\_15 DEL\_DEP15 OAG\_ARPT\_DEP\_15\_PCNT PTM\_ARPT\_DEP\_15\_PCNT T\_DIF\_G2G\_A OAG\_ARPT\_DEP\_15 PTM\_ARPT\_DEP\_15 T\_DIF\_G2G OAG\_ARPT\_DEP\_15C EDCT\_DEP\_LATE DEP\_SCORE T\_DEP\_EDCT\_HOLD DEP\_RATE VISIBLE CRSDepTime DEP\_EDCT\_CNT DEP\_CT Origin\_Demand DEP\_DEMAND MC T\_DELAY\_TO DEP\_CNT DayOfWeek CRSElapsedTime EDCT\_DEP\_EARLY DEL\_TAXI\_OUT\_CNT DELAIR\_CT\_0 OAG\_DEP DayofMonth UniqueCarrier TEMP CEILING Month Year

view as

1 0.940341 0.519114 0.505591 0.501095 0.496224 0.485998 0.480119 0.475827 0.47486 0.459773 0.455583 0.436198 0.429131 0.423596 0.381933 0.364813 0.356197 0.345199 0.337274 0.336702 0.33172 0.325128 0.301155 0.260843 0.208704 0.195301 0.19333 0.183316 0.14456 0.129106 0.124254 0.115188 0.114943 0.109249 0.100847 0.084915 0.062214 0.06197 0.049895 0.040588 0.034901 0.023255 0.021332 0.010642 0.010048 0.008393 0.005223 0.000279 0

Spearmen Correlation

DepDelay totaldelay T\_O\_GATE\_DELAY\_A T\_GATE\_DELAY\_A O\_GATE\_DEL\_15\_PCNT DEL\_DEP15\_PCNT T\_O\_GATE\_DELAY T\_OAG\_ARPT\_DEP\_A T\_GATE\_DELAY T\_PTM\_ARPT\_DEP\_A O\_GATE\_DEL\_15C OAG\_ARPT\_DEP\_15\_A OAG\_ARPT\_DEP\_15M OAG\_ARPT\_DEP\_15\_PCNT O\_GATE\_DEL\_15 T\_OAG\_ARPT\_DEP PTM\_ARPT\_DEP\_15\_PCNT DEL\_DEP15 OAG\_ARPT\_DEP\_15 PTM\_ARPT\_DEP\_15 T\_ARR\_EDCT\_HOLD O\_GATE\_DEL\_15\_A OAG\_ARPT\_DEP\_15C T\_DIF\_G2G\_A T\_DIF\_G2G EDCT\_DEP\_LATE DEP\_RATE T\_DEP\_EDCT\_HOLD CRSDepTime VISIBLE DEP\_SCORE DEP\_EDCT\_CNT MC Origin\_Demand DEP\_DEMAND T\_DELAY\_TO DEP\_CT EDCT\_DEP\_EARLY DayOfWeek CEILING UniqueCarrier CRSElapsedTime DELAIR\_CT\_0 DEP\_CNT DayofMonth TEMP DEL\_TAXI\_OUT\_CNT OAG\_DEP Month Year

view as

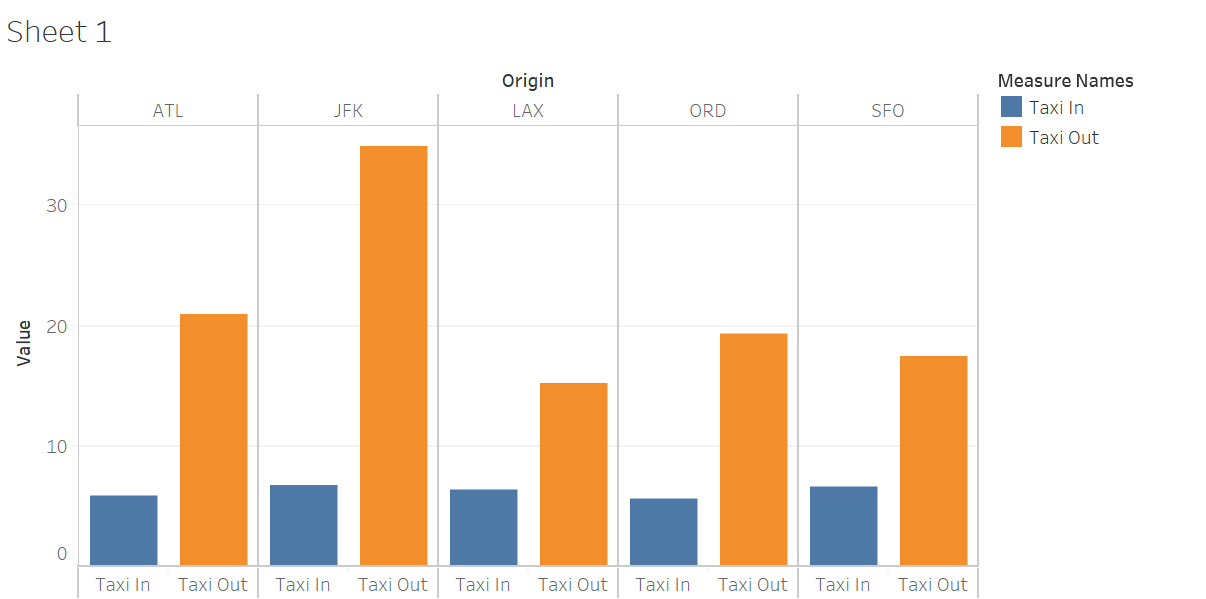
1 0.8135 0.392558 0.39031 0.390083 0.387044 0.368194 0.366677 0.364085 0.359288 0.354539 0.349801 0.337617 0.329012 0.326783 0.326243 0.325019 0.322596 0.318652 0.3151 0.300711 0.282006 0.263787 0.251163 0.220906 0.186549 0.165979 0.158368 0.152016 0.135931 0.127672 0.126946 0.114183 0.076799 0.075524 0.070372 0.063592 0.052883 0.052488 0.050222 0.042204 0.038884 0.035984 0.035007 0.011689 0.011226 0.009355 0.00923 0.000461 0

* The weather data was not showing effects on the overall variable selection process as we expected it to show. We have taken into consideration certain weather features while predicting the number of delays but haven’t done so while classification as we were getting better accuracy scores while classification without weather data.
* We decided to go with the Airport based models for both classification and prediction which will predict the delays based on the Fixed departure airport ATL (Atlanta Hartsfield Jackson Intl. Airport) and We will see further why we chose ATL during EDA part, the airport based model in said to give out maximum accuracy per the reports. The report which is referred to is available at: <https://srcole.github.io/assets/flight_delay/report.pdf>.

1. EDA (Exploratory Data Analysis):

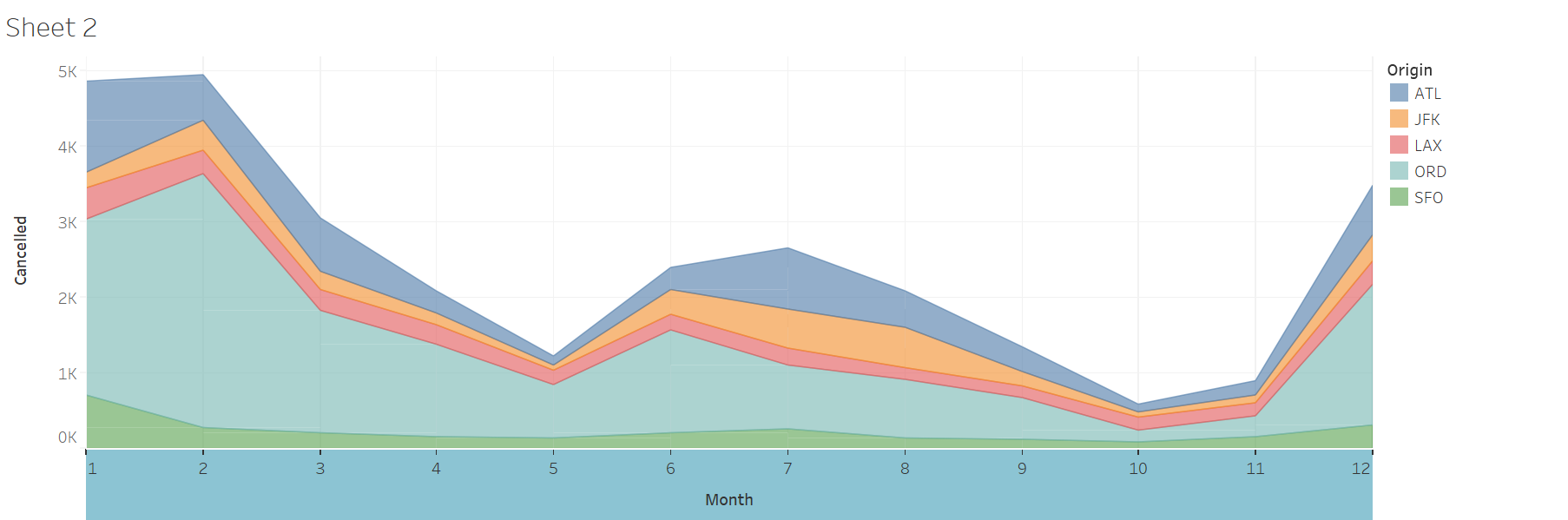
We have tried to incorporate both tableau and python while doing data analysis. The first four sheets have been published on Tableau public and rest are python inferences of EDA.

* *Taxi-Out Time:* The time elapsed between departure from the origin airport gate and wheels off.
* *Taxi-In Time:* The time elapsed between wheels-on and gate arrival at the destination airport.

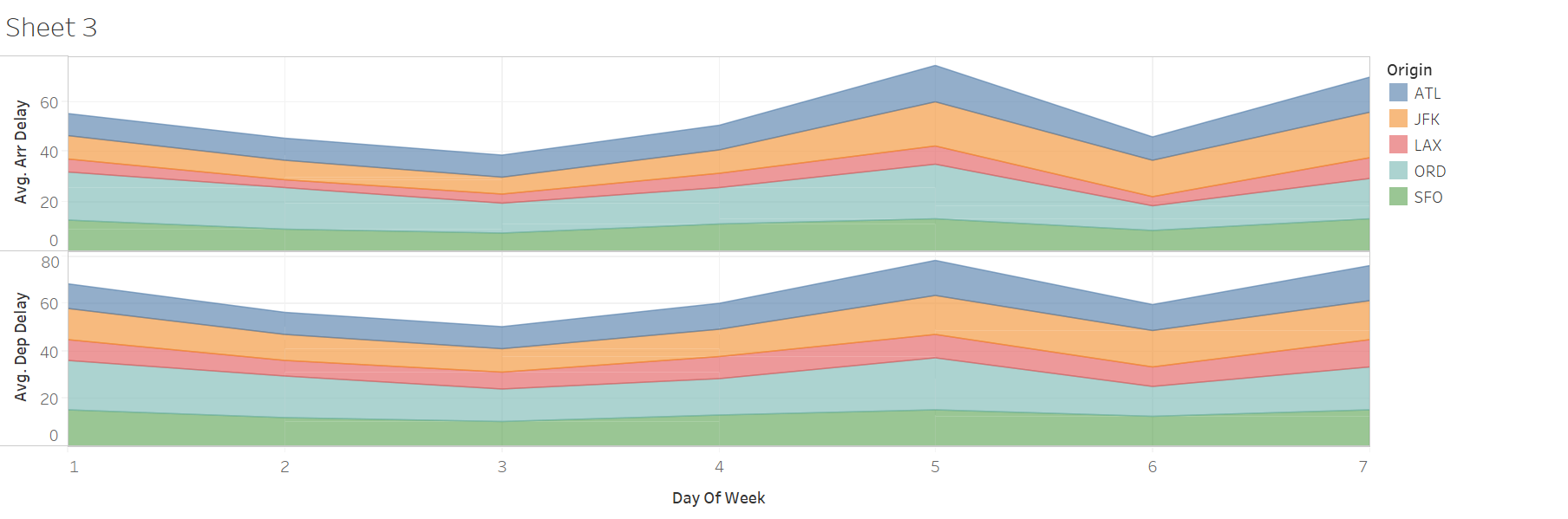


This factor is giving interesting insights in the delays happening across the major US airports.

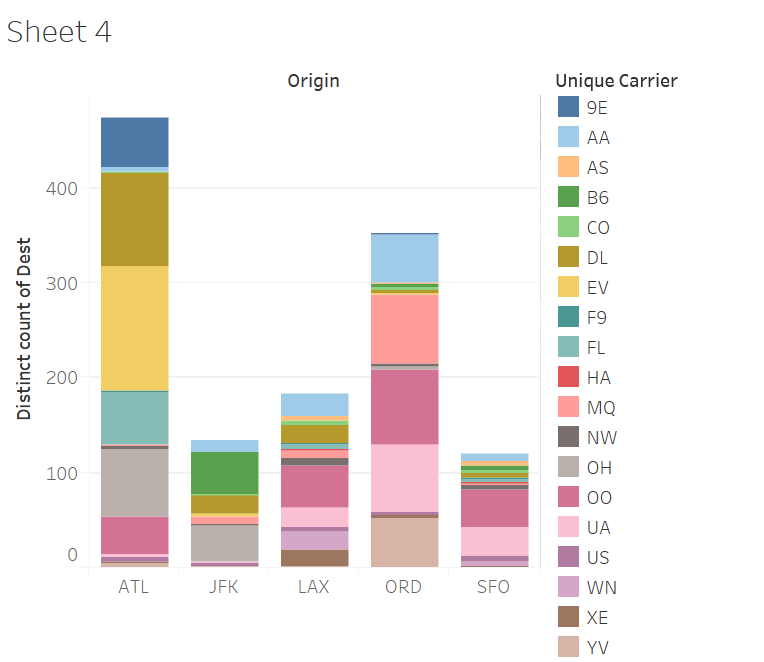
Let’s check out the cancellation patterns occurring over months at major US airports.



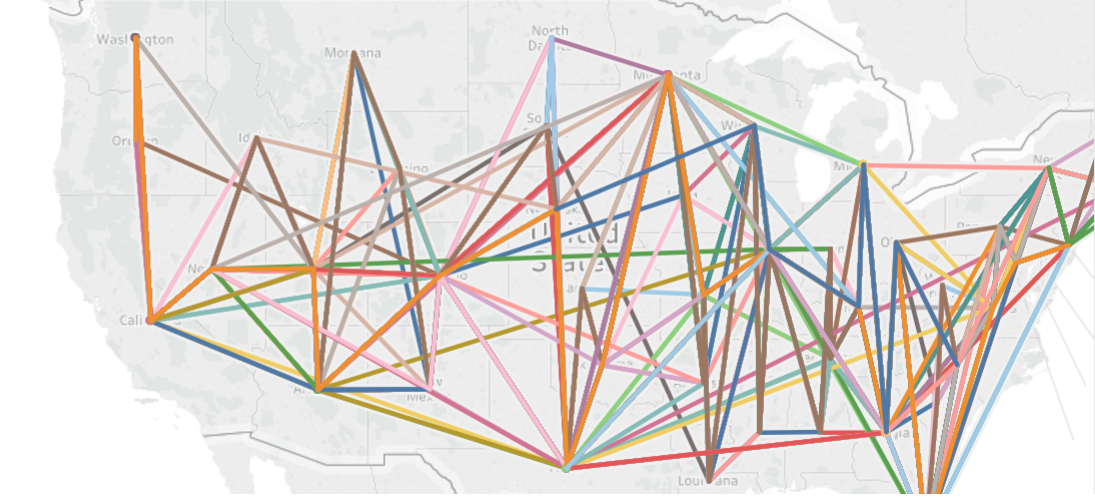
The trend of average Arrival delay and departure delay can be seen from this graph.



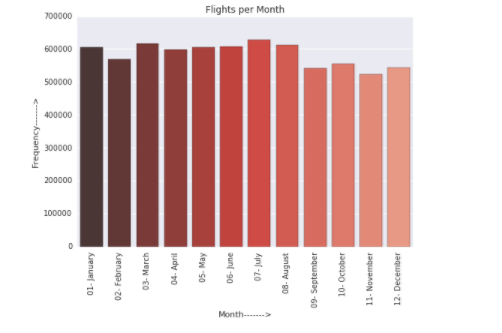
Multiple carriers across Major American airports:



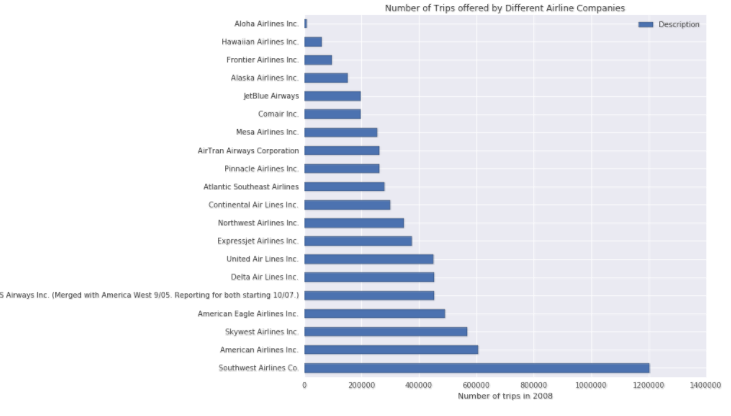
The various routes of flights across US



Maximum flights taken across months can be seen in this chart



The number of flights offered by various carriers across US can be seen in this chart.



1. Machine learning:

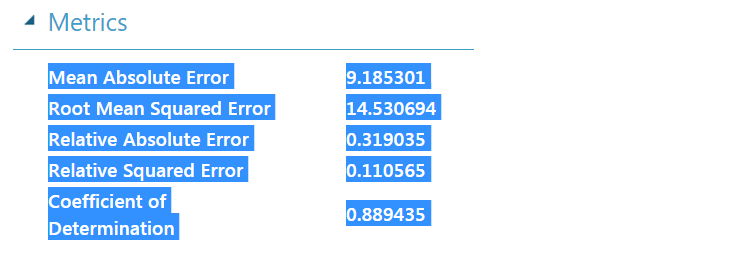
The data is nonlinear in nature which means it won’t be able to apply linear algorithms to classify the data the way we wish to make it classify, Also the data is highly unbalanced making it very tough to either do prediction or classification

* + - Prediction:

We have tried many combination while performing the prediction of delays in minutes based on the given input parameters

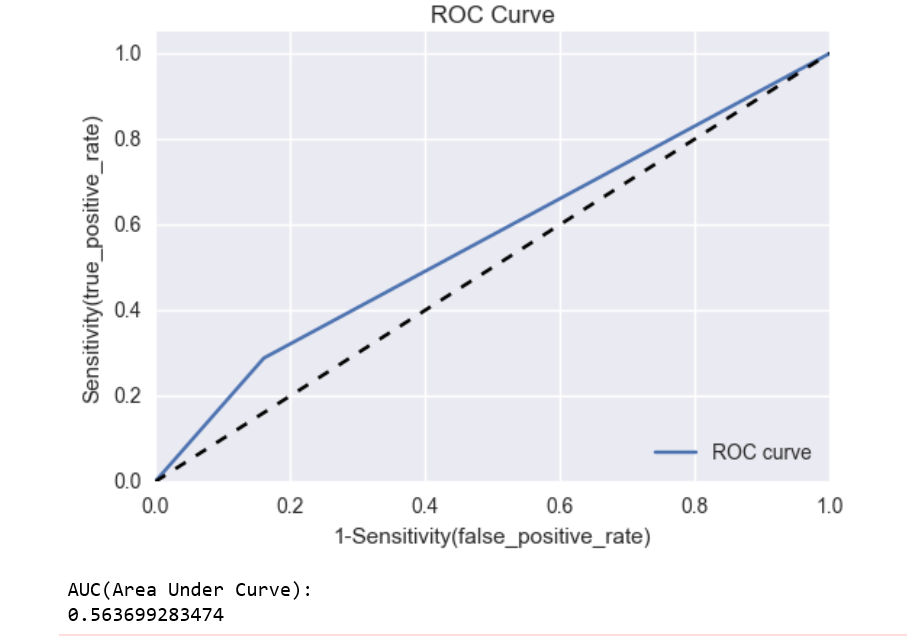
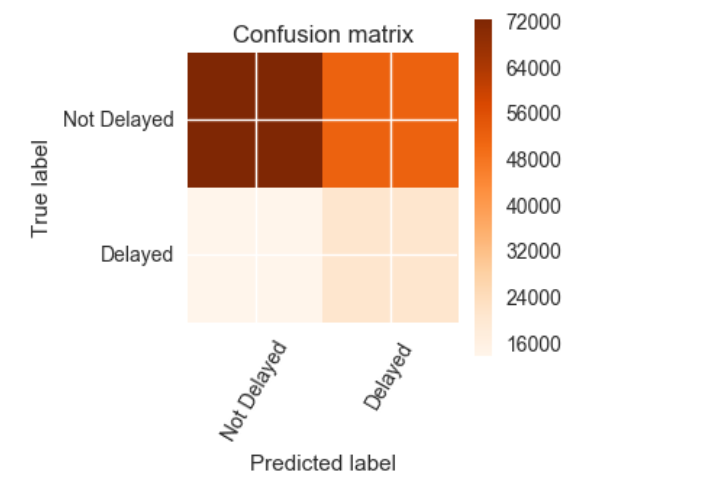
Our approach included: ( Model type: Tree based Random forests)

* + - * Taking overall flights data of 8 million records which we know won’t give predictions because of this huge dataset which was not possible on IBM bluemix as well as our local machines (Dell inspiron Intel i7)
      * Then we tried going for subsampled data taking random 800,000 records and min max scaling it and equaling the number of delayed and non-delayed flights which also gave out very bad results.
      * Finally, we took a specific airport ATL as base airport and developed a airport based model which in past has proven to be effective which worked for us also and gave out error of 9 minutes, which is not as bad as achieving above 1000 minutes of error in previous models.

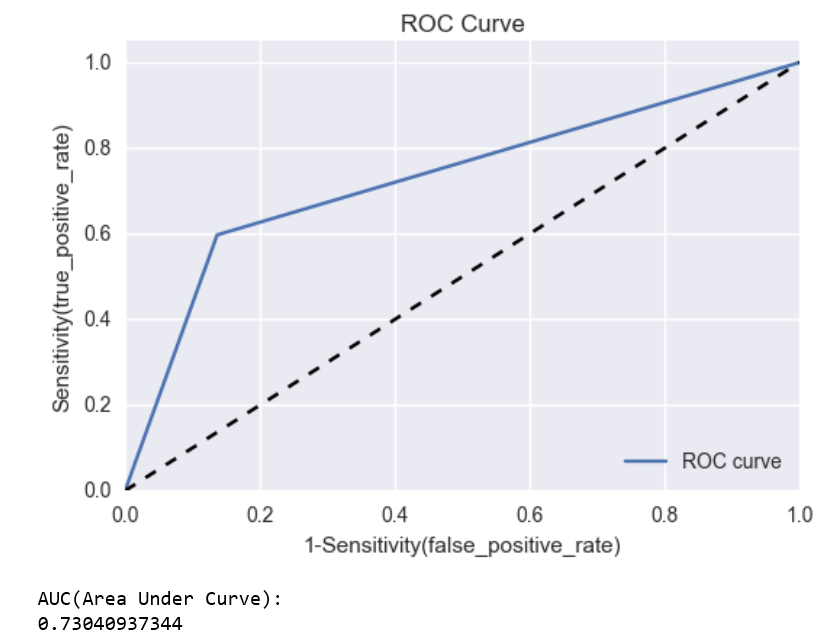
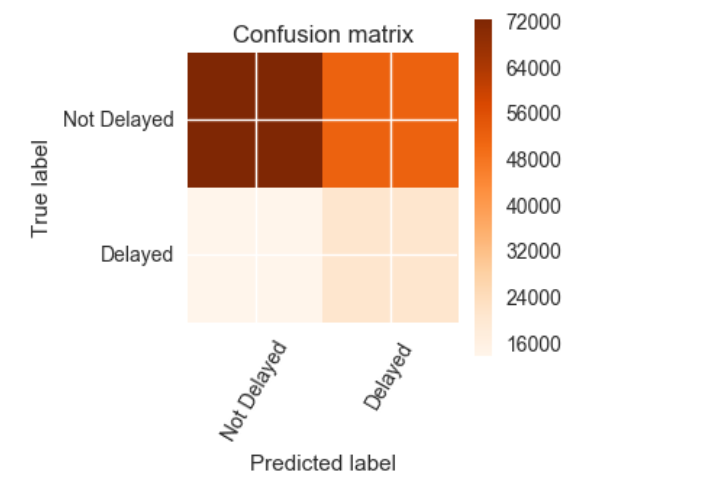


* + - Classification:

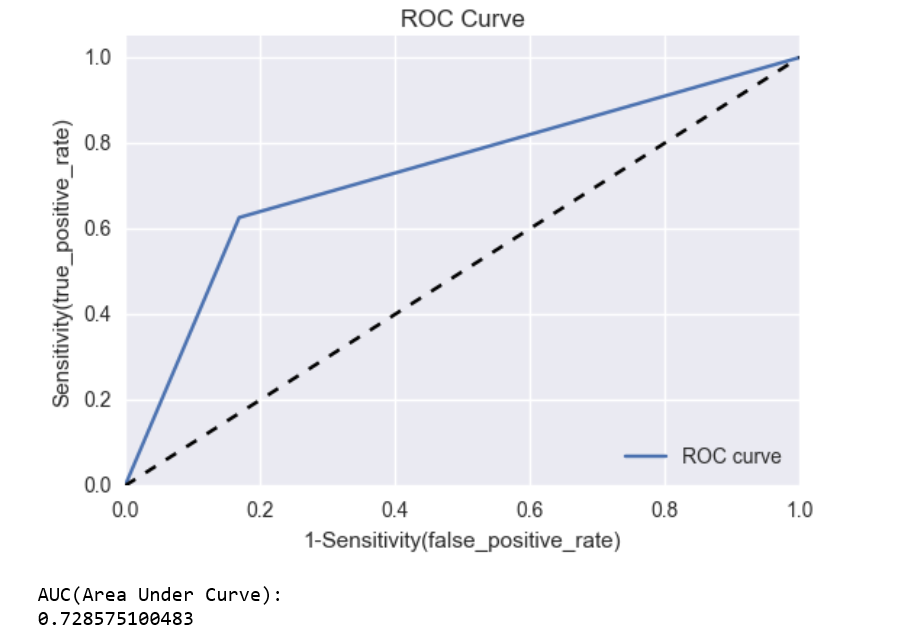
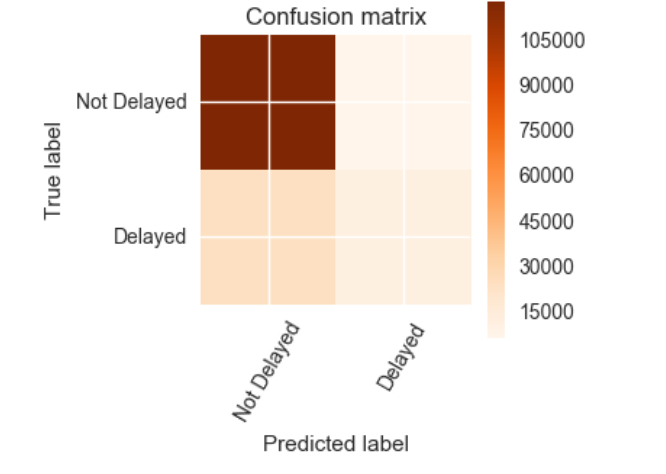
1. Logistic Regression:



1. Random Forest:



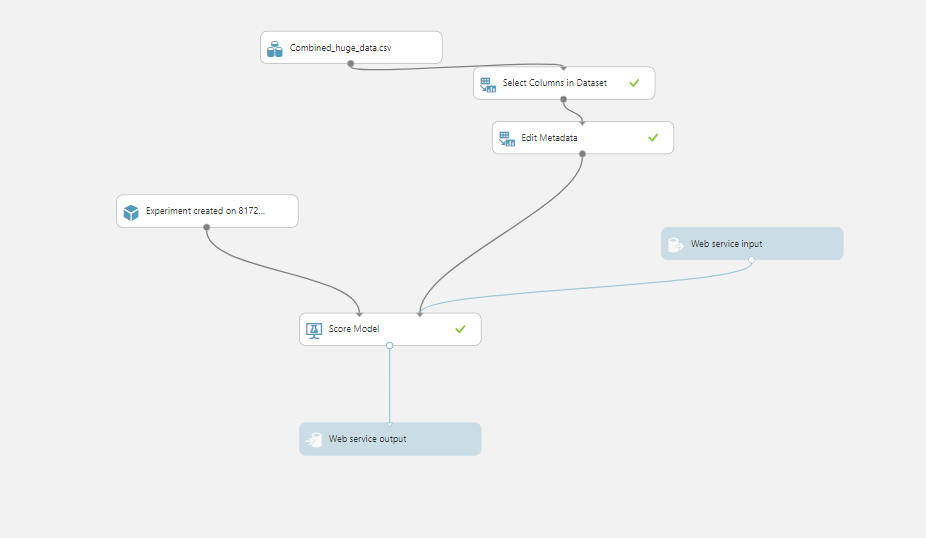
1. KNN (K nearest Neighbour classifier):



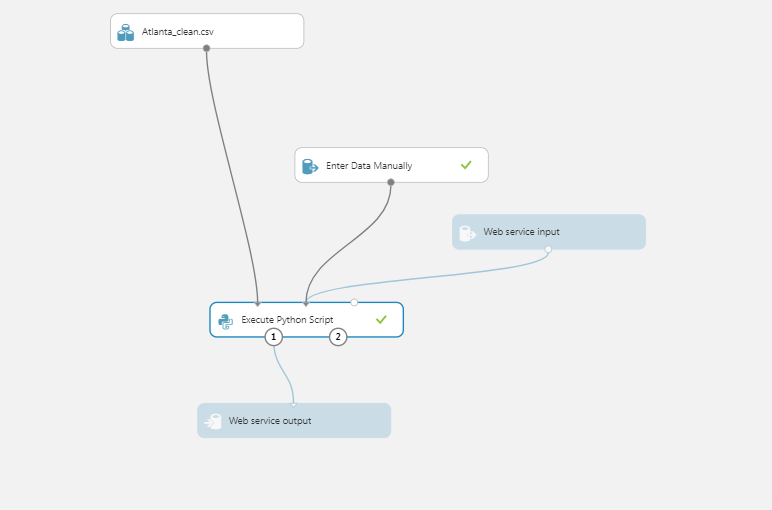
As we can see the tree based algorithm Random Forest out performs other algorithms having maximum **AUC of 0.7304**

Deployed on cloud using AZURE ml:

* + - * + We have used our own python scripts on the azure web ML.
* Prediction:

****

* Classification:

****

# **Web App**





DOCKER:

docker pull sumedh11/team4\_final

docker run sumedh11/team4\_final

*Some points worth concluding and observing:*

1. Due to unbalance dataset, We have to run tree based algorithms or SVM to achieve a good model, The problem with SVM is that it takes very long time and lot of computational power to train, We tried running it on IBM blue mix as well as local and kept it running for almost 2 days but were unable to achieve any success due to large amount of data almost 800,000 records, then we tried another approach of subsampling the data and making equal records of non-delayed and delayed and let it run for 6 hours but due to time constraints could not go further, which leaves with only one option that is tree based algorithm to give us good models.
2. As we can see the tree based Random Forests have turned out to be the best model giving us AUC (Area under curve) of 0.73040937344 which outperforms other Logistic and KNN.
3. Also, it is proven that Tree based algorithms are said to perform very well on the categorical dataset, which fits perfect for this case.
4. The Random Forest model was highly successful in predicting which flights were not delayed giving out more than 90% accuracy, and less successful at predicting delayed flights of over 40% delayed flights correctly predicted.