FLIGHT DELAY PREDICTION

Developed by: Aaqib Ali

AGENDA

- Introduction
- Data Analysis and Exploration
- Factors and Causes
- Model Approach
- Model Evaluation Metrics
- Model Performance and Results
- Pipeline Architecture
- Further Improvements
- Future Direction

INTRODUCTION

- In this task I have to analyse the given “Airline On-Time Performance” dataset and build a predictive model that can predict the flight delay. Moreover, highlight some of the root causes of the flight delays.
- In addition, how we can enhance the predictive power of the implemented model and how these flight delays could be avoided or minimized.
- Given dataset was provided by “Bureau of Transportation Statistics” from year 1987 to 2012.

DATA ANALYSIS AND EXPLORATION

- Data exploration and analysis is carried out into 7 different directions so that can we have a better
  understanding about causes of flight delays.

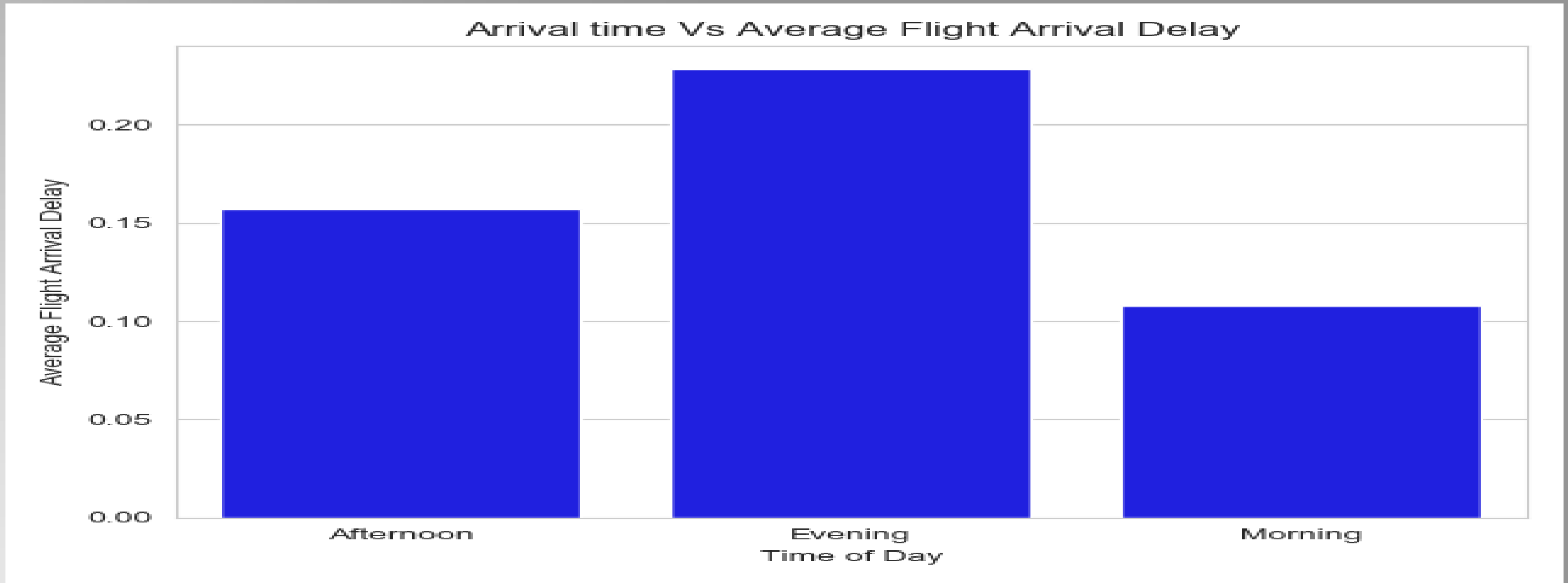
- These 7 directions helps us to explore the rare events of flight delays.

- These directions are:
  - Time of the day vs flight delays
  - Flight route vs flight delays
  - Flight origin and state vs flight delays
  - Flight carrier vs flight delays
  - Flight date vs flight delays
  - Part of week vs flights delays
  - Number of delays

DATA ANALYSIS AND EXPLORATION

Time of the day vs flight delays:

- Below graph demonstrates that the number of flights arrive delay against the time of the day.

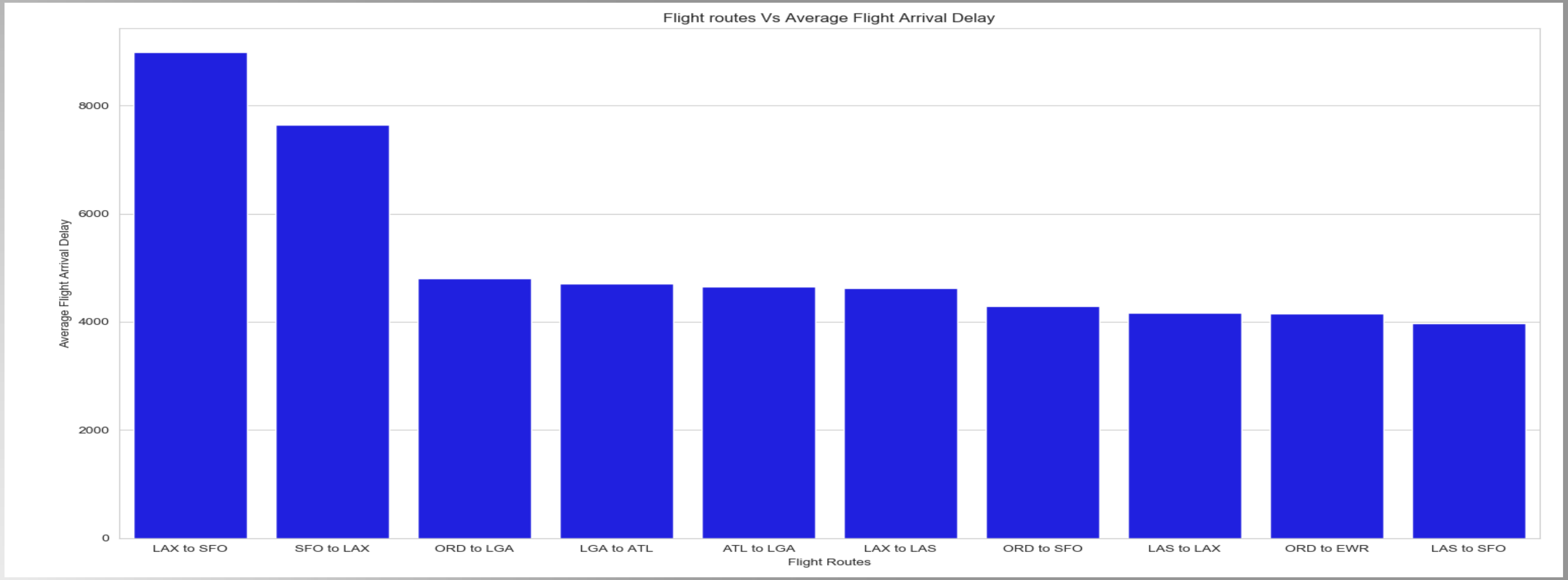


- We observe quite high flights delay during evening time.

DATA ANALYSIS AND EXPLORATION

Flight route vs flight delays:

- Below graph demonstrates that the number of flights arrive delay against the route of a flight.

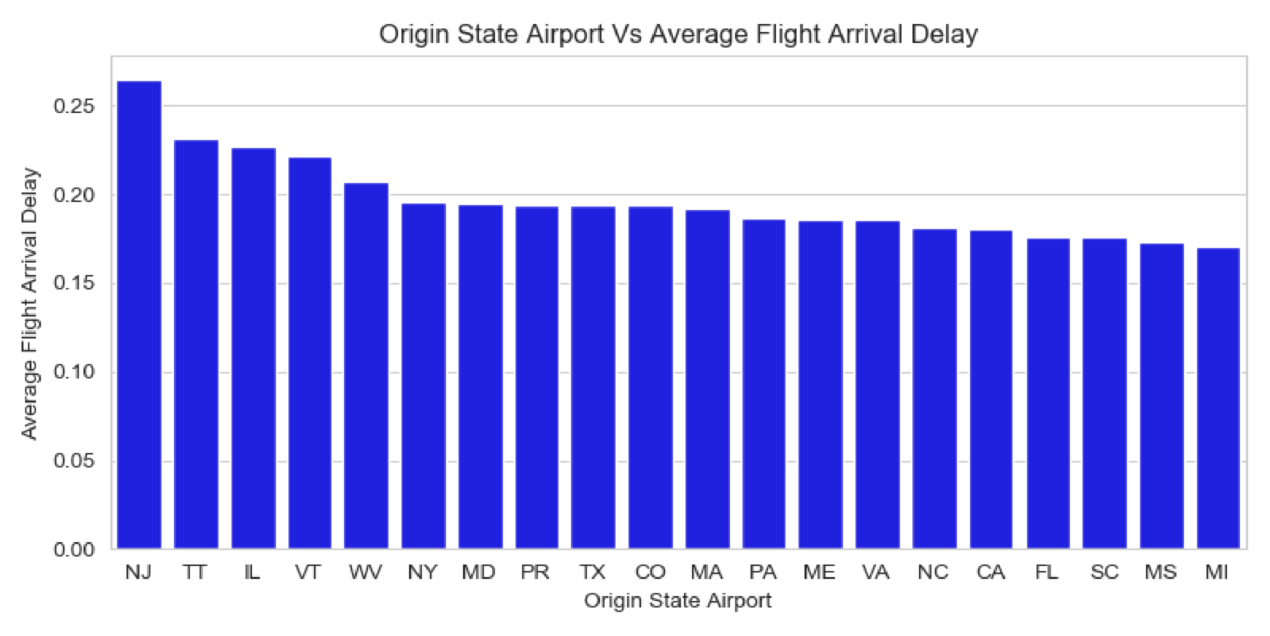


- Route LAX to SFO and vice versa caused more number of flight delays.

DATA ANALYSIS AND EXPLORATION

Flight origin and state vs flight delays:

- Below graph demonstrates that the number of flights arrive delay against the origin state of a flight.

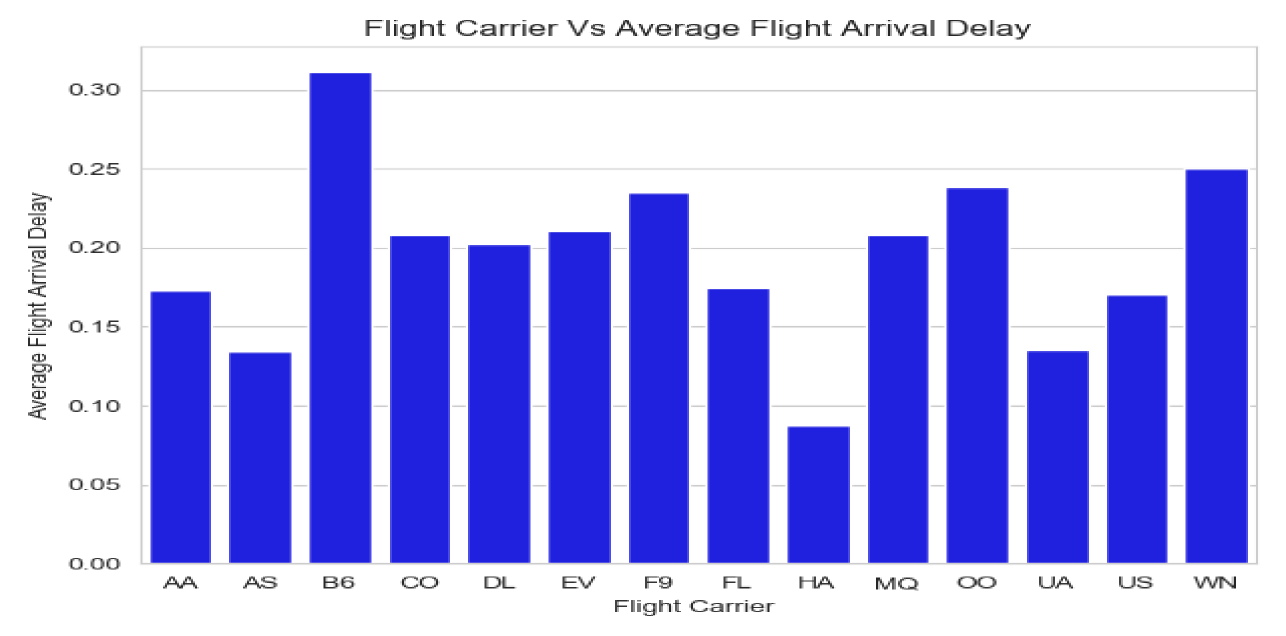


- NJ and TT airport seems to be more busy with high number of flight delays.

DATA ANALYSIS AND EXPLORATION

Flight carrier vs flight delays:

- Below graph demonstrates that the number of flights arrive delay against the flight carrier.

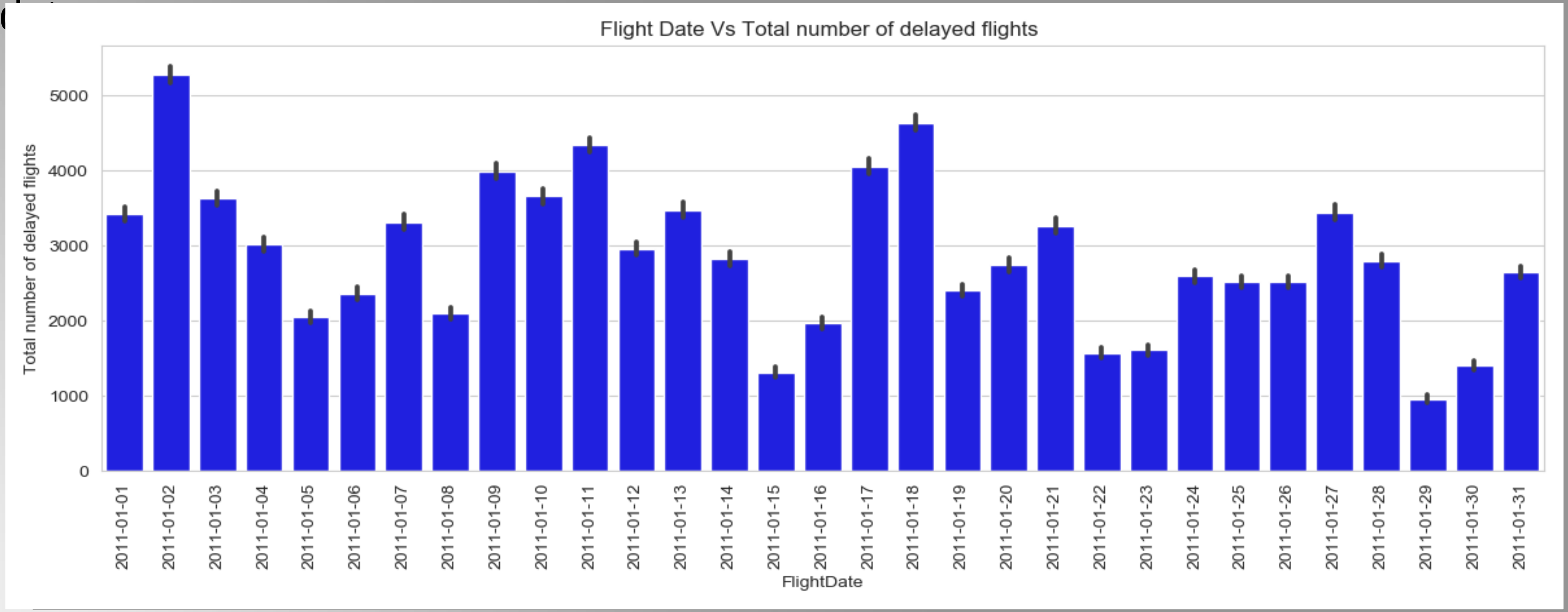


- Flight carrier B6 causes more number flight delays as compared to others.

DATA ANALYSIS AND EXPLORATION

Flight date vs flight delays:

- Below graph demonstrates that the number of flights arrive delay against the flight d

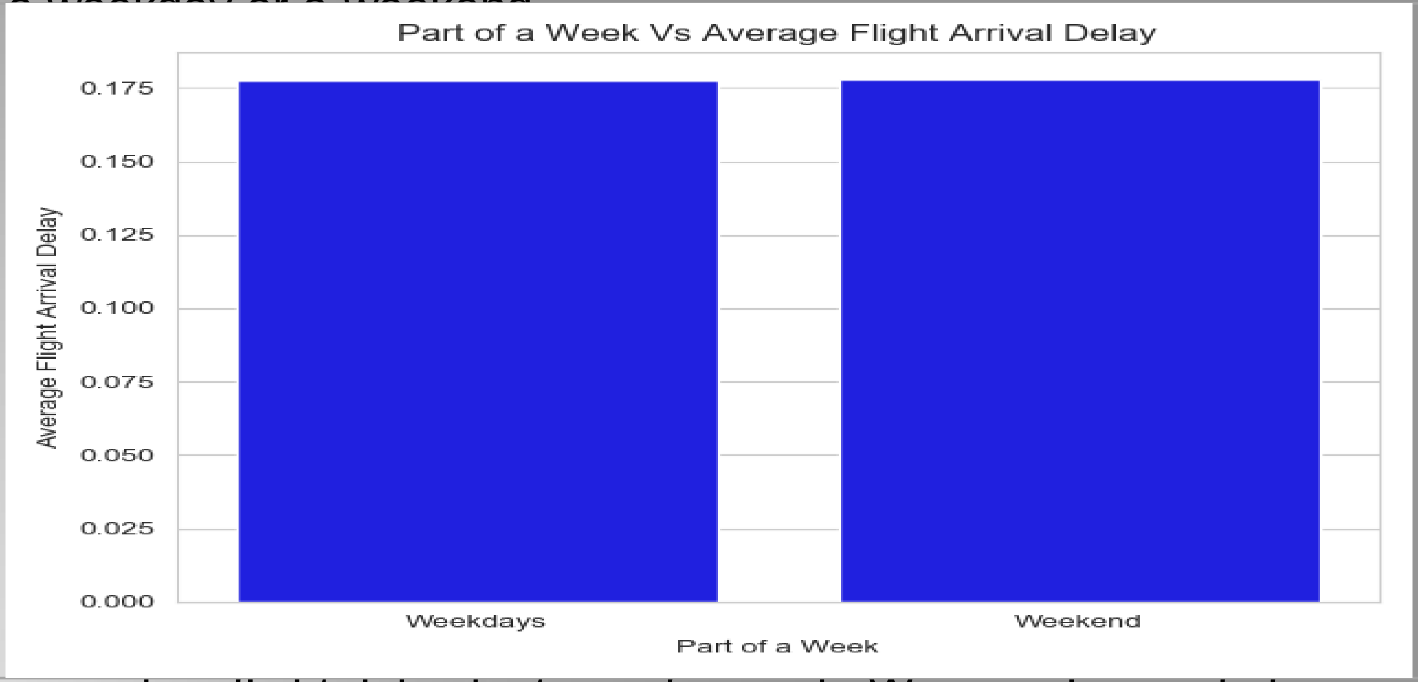


- Common pattern is observed for number of flights delay per day except for 2/01/2011 and 17/01/2011.

DATA ANALYSIS AND EXPLORATION

Part of week vs flights delays:

- Below graph demonstrates that the number of flights arrive delay against the part of the week
whether it's a weekday or a weekend.

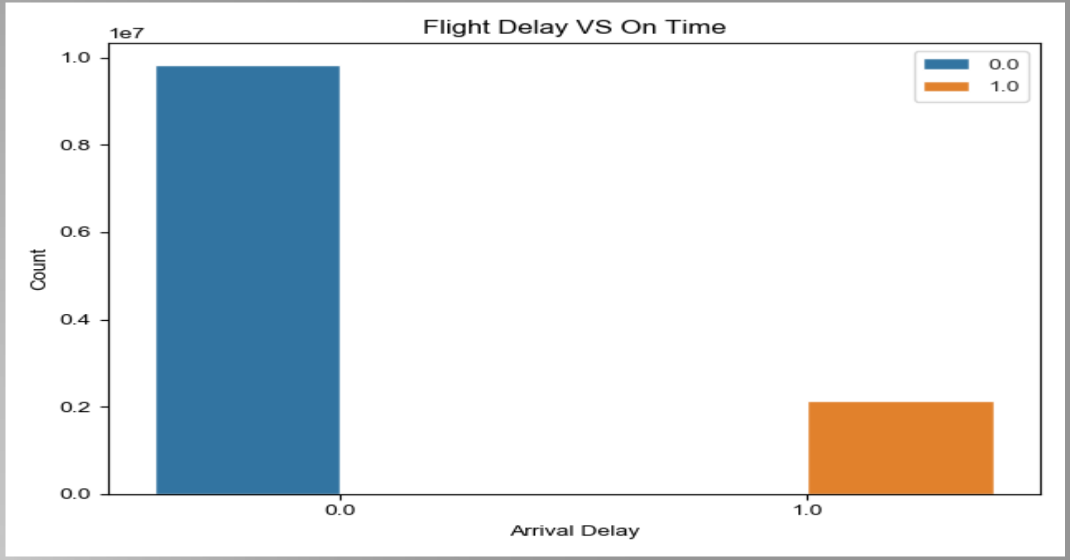


- No effect observed on flight delay by time of a week. We can also verify from our previous graph as 02/01/2011 is a weekday and 17/01/2011 is a weekend.

DATA ANALYSIS AND EXPLORATION

Number of delays:

- Below graph demonstrates that the number of flight delays are less then the number of flight on time. As, flight delay is a rare event, that leads us to a common Machine Learning problem of data imbalance.



- Where “0” demonstrates flight on time and “1” flight delayed

FACTORS AND CAUSES

- After our extensive data analysis and exploration I came to the final conclusion that there are some of the factors and causes that are involved in flight delays.
- These factors and causes can be listed as:
  - Around 60% of the flights are delayed during evening or afternoon. As there are more number of flights operated during this time that cause the flight delay the reason could be during this time there is much more traffic on runways that leads to the flight delays.
  - Flights from LAX to SFO and vice versa tends to be delayed frequently, the reason could be long distance to cover by the flight.
  - NJ and TT state airports causes more number of flight delays, the reason could be busy airport or a security checks
  - Flight Carrier B6 causes more flight delays, the reason could be the number of planes operated are less than number of routes.
  - Data exploration also demonstrates that long taxi of a flight also leads to flight delays, the reason could be big airports where runways are far away from the parking.

MODEL APPROACH

- SGDClassifier is used as a baseline model. The performance is improved by using tree based models
  i.e. Random Forest and XGBoost.

- Random Forest and XGBoost are highly interpretable and easy to understand as compared to regression based models.

- If you want to explain the model performance and working to the business people, then these tree based models are the best option to work with. This is the reason I am working with these models.

- I achieved best performance and accuracy with XGBoost while SGDClassifier performs quite low.

- To deal with the data imbalance problem I used a weighted loss technique which is considered to be the most effective Machine Learning technique as compared to down sampling and over sampling.
- For features selection, I extensively used feature importance, panda profile report and L1 regularization (Lasso Technique).

MODEL EVALUATION METRICS

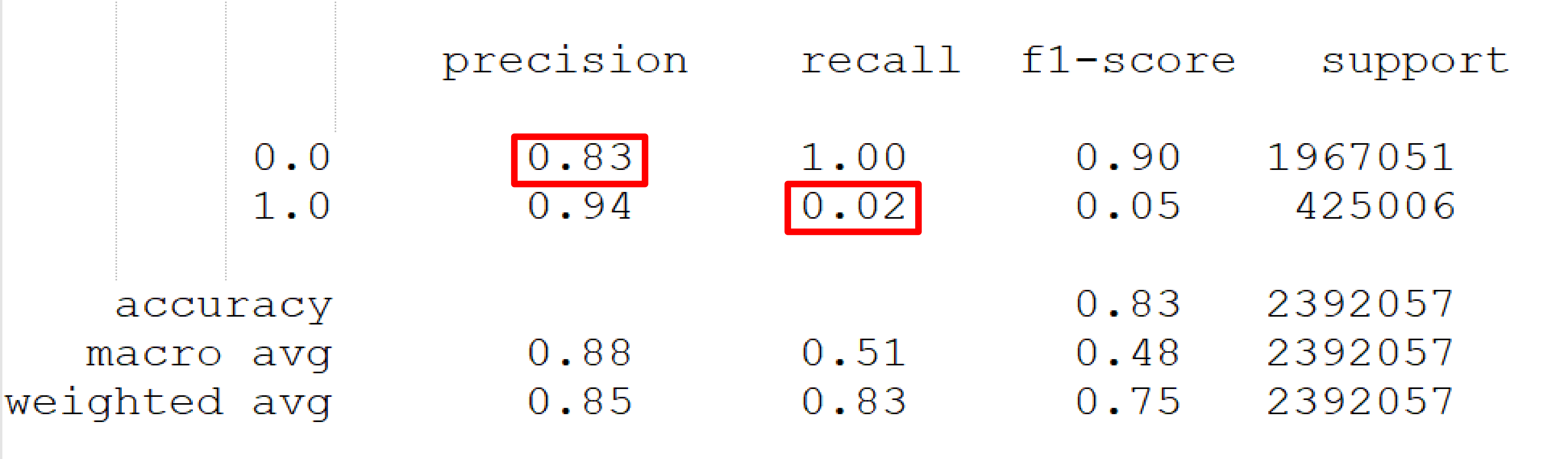
- As, we are dealing with the supervised classification problem, for this type of problems "Accuracy" is the most suitable evaluation metrics.
- However, we have a data imbalance problem, in our case "Accuracy" metrics will not work.
- We will get 90% accuracy as our model is predicting negative class every time and unable to classify the Positive class at all. Then this 90% accuracy is actually wrong.
- For our special case, I used "Area under the curve" metrics which is the suitable metrics for imbalance datasets.
- Our main goal is to Improve the "Precision" of the negative class (Flight on Time) as well as "Recall" of the positive class (Flight Delayed).

MODEL PERFORMANCE AND RESULTS

- XGBoost is trained on 80% of the final dataset and 20% of the dataset was used to validate/test the
  performance of the model.
- I also split data on yearly basis as we use to do in professional environment. i.e. I trained my model on January 2011 to June 2012 and validate/test the performance on July 2012 to December 2012.
- Trained SGDClassifier, Random Forest and XGBoost. However, final submission contains the best performed model i.e. XGBoost.
- All the model hyper parameters are also tuned and final model is trained on the best parameters.
- Final result contains ROC curve, Confusion Matrix and the classification report.

MODEL PERFORMANCE AND RESULTS

SDG Classifier with Imbalance dataset:

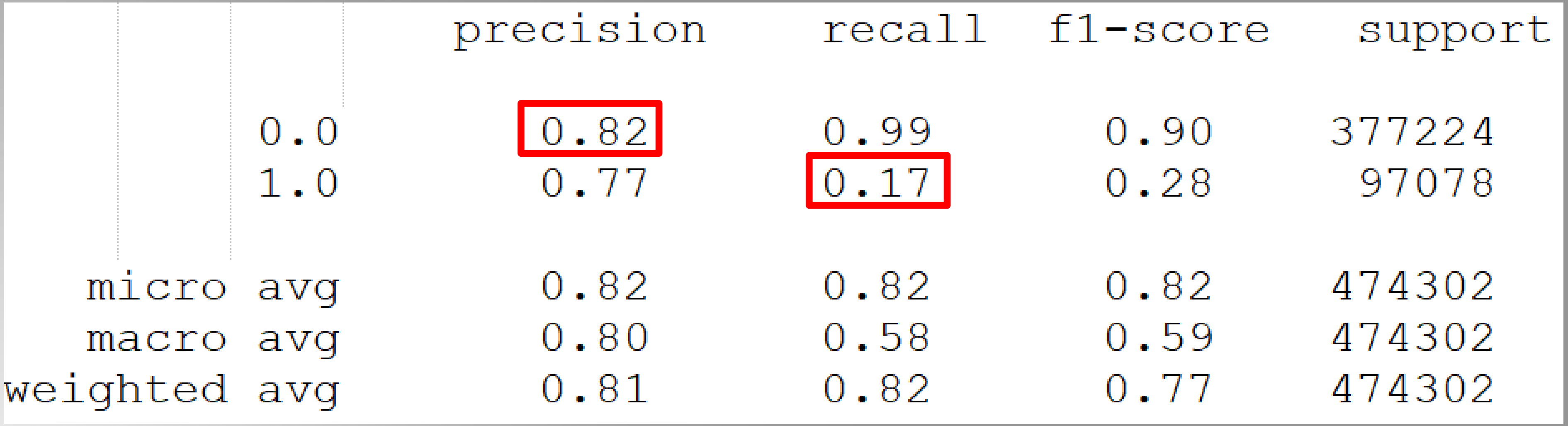


```
              precision    recall  f1-score   support

         0.0       0.83      1.00      0.90   1967051
         1.0       0.94      0.02      0.05    425006

    accuracy                           0.83   2392057
   macro avg       0.88      0.51      0.48   2392057
weighted avg       0.85      0.83      0.75   2392057
```

- We can clearly see that on imbalance dataset, model is not able to classify flight delay events as we are getting very low recall of 0.02.

MODEL PERFORMANCE AND RESULTS

SDG Classifier with balance dataset:

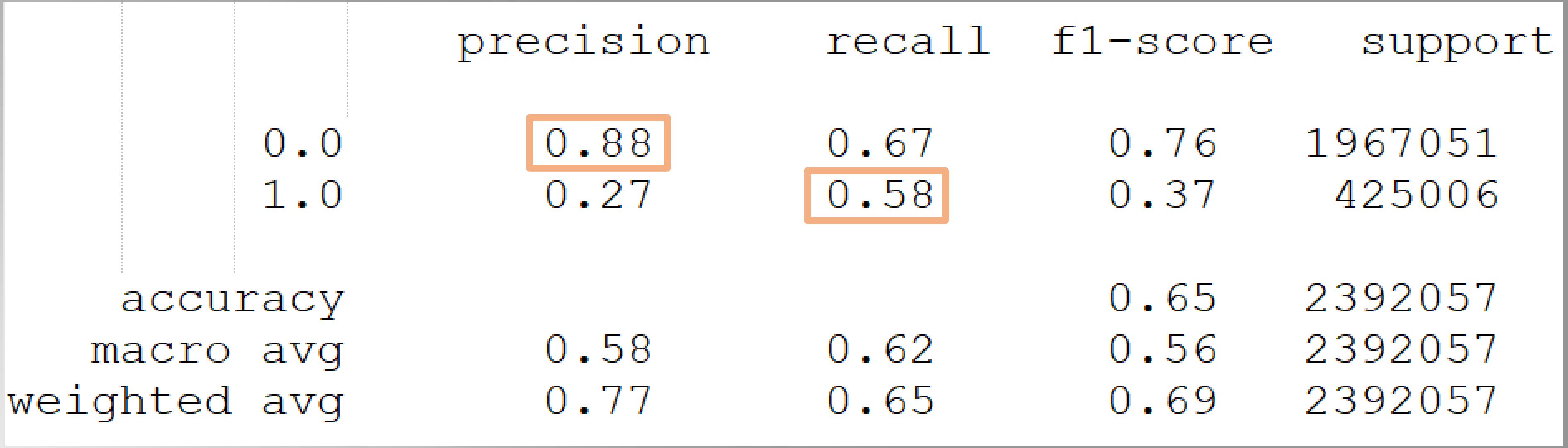


```
              precision    recall  f1-score   support

         0.0       0.82      0.99      0.90    377224
         1.0       0.77      0.17      0.28     97078

   micro avg       0.82      0.82      0.82    474302
   macro avg       0.80      0.58      0.59    474302
weighted avg       0.81      0.82      0.77    474302
```

- After handling imbalance dataset, now model is able to classify flight delays. However, we still got quite low recall of 0.17.

MODEL PERFORMANCE AND RESULTS

Random Forest with balance dataset:

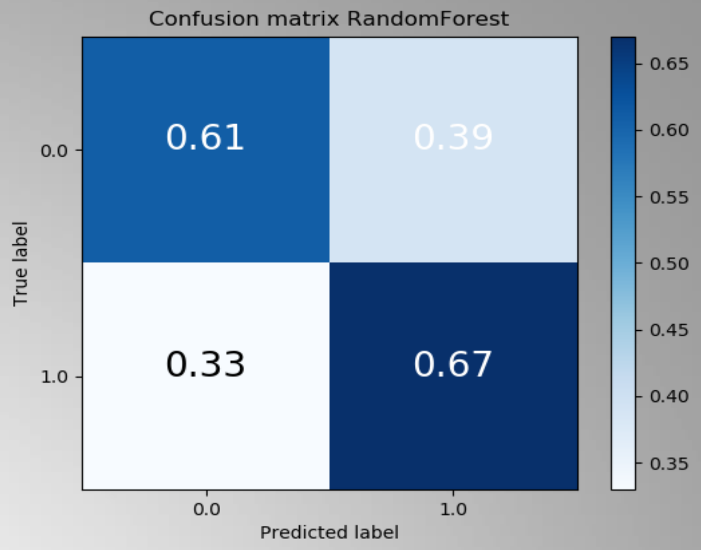


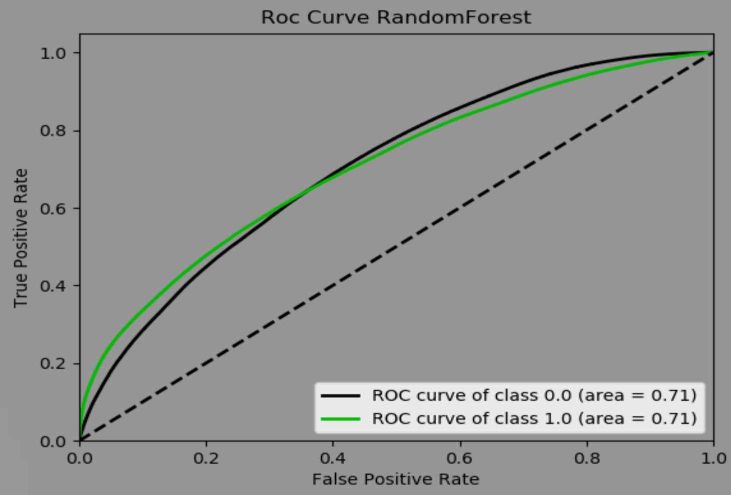
```
              precision    recall  f1-score   support

         0.0       0.88      0.67      0.76   1967051
         1.0       0.27      0.58      0.37    425006

    accuracy                           0.65   2392057
   macro avg       0.58      0.62      0.56   2392057
weighted avg       0.77      0.65      0.69   2392057
```

- Random forest is performing very well, we achieve recall of 0.58. However, we will try to improve it more.

MODEL PERFORMANCE AND RESULTS

Random Forest with balance dataset:

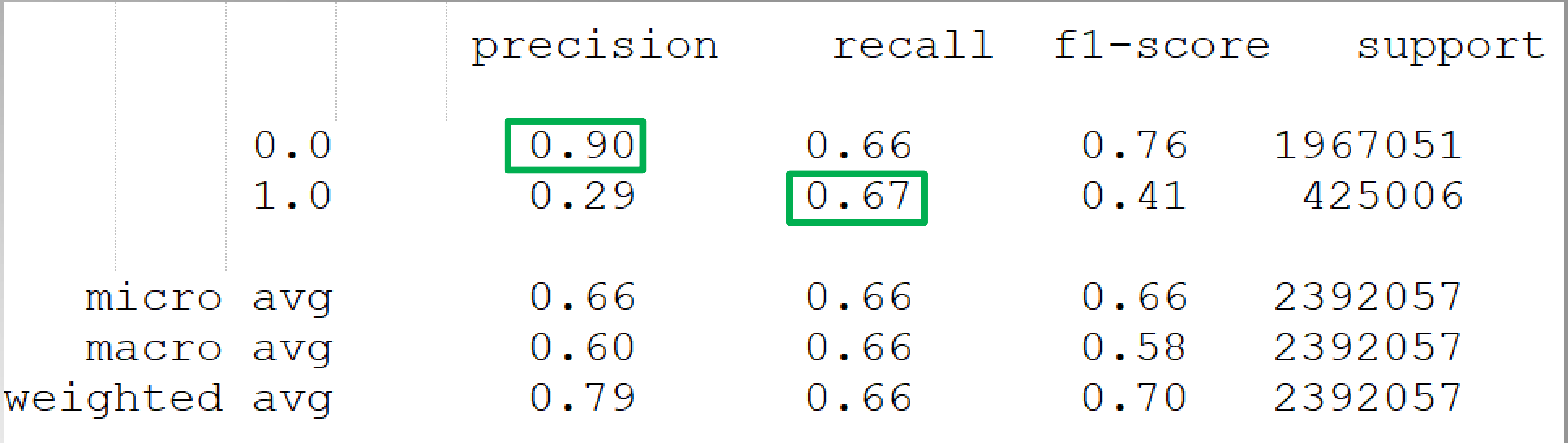




- Random forest is covering 71% of the area under the curve (AUC) and also predicting both classes perfectly. However, some of the negative labels are wrongly classified which needs to be improve.

MODEL PERFORMANCE AND RESULTS

XGBoost with balance dataset:

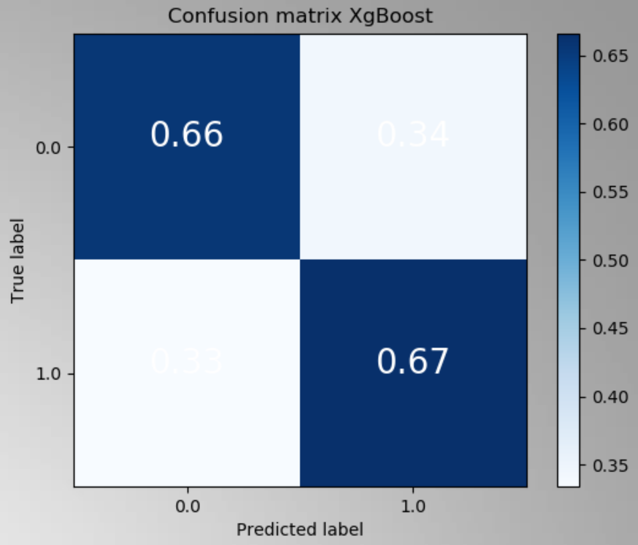


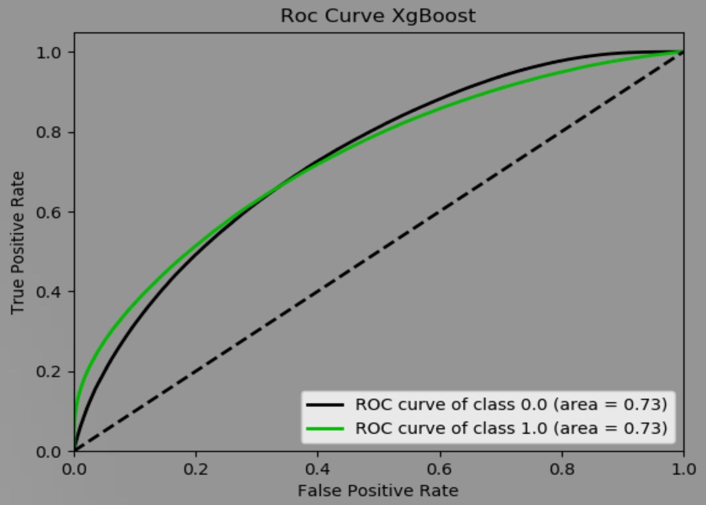
```
              precision    recall  f1-score   support

         0.0       0.90      0.66      0.76   1967051
         1.0       0.29      0.67      0.41    425006

   micro avg       0.66      0.66      0.66   2392057
   macro avg       0.60      0.66      0.58   2392057
weighted avg       0.79      0.66      0.70   2392057
```

- XGBoost even improves more and achieves the recall of 0.67. Moreover, it also improves the precision of the positive class.

MODEL PERFORMANCE AND RESULTS

XGBoost with balance dataset:

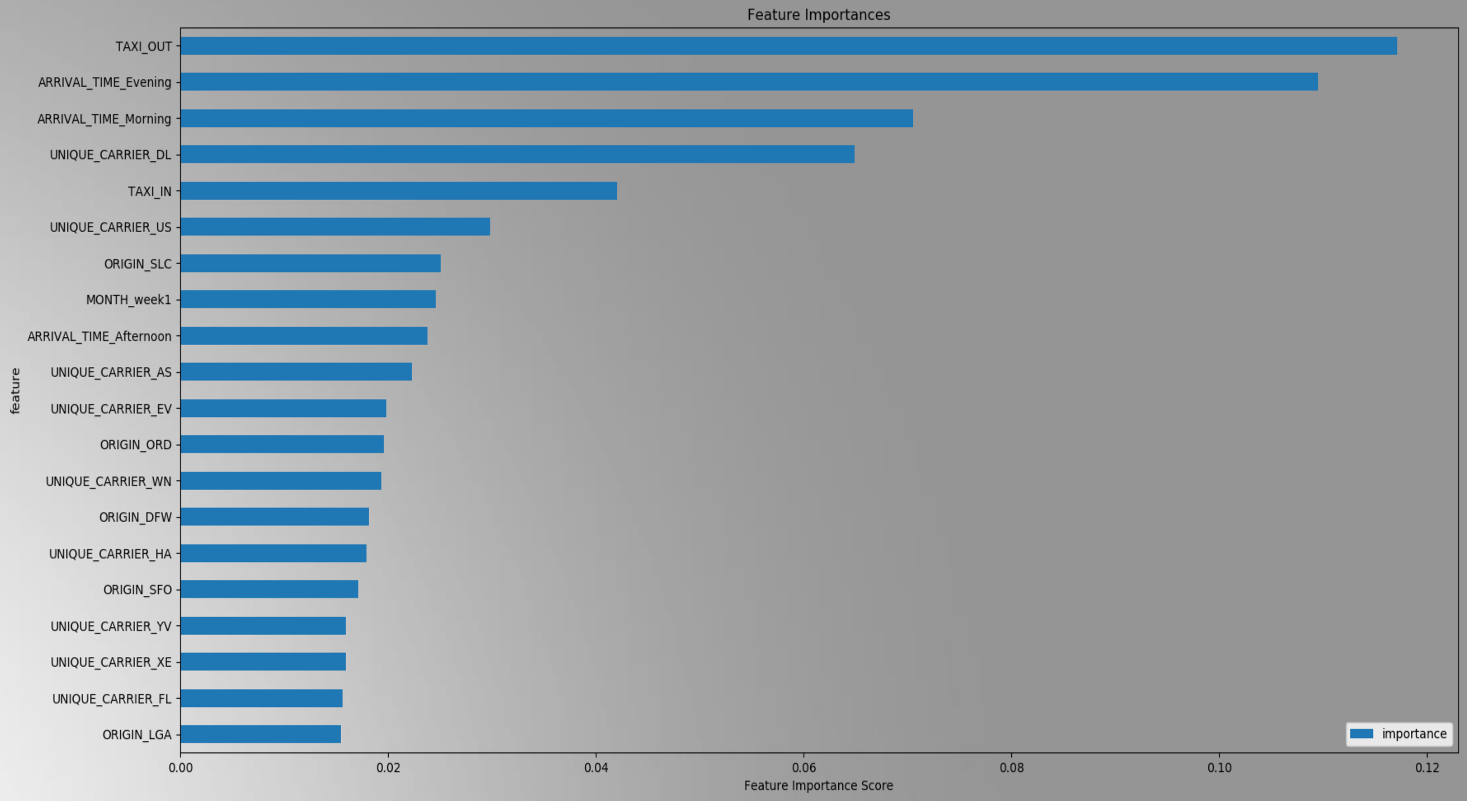




- XGBoost is performing very well on the dataset by covering 73% of the area under the curve. Moreover, model is predicting both classes perfectly.

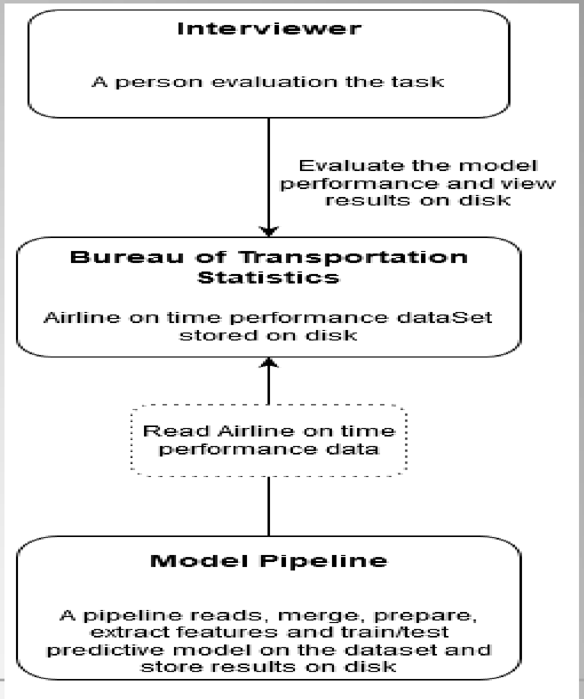
MODEL PERFORMANCE AND RESULTS

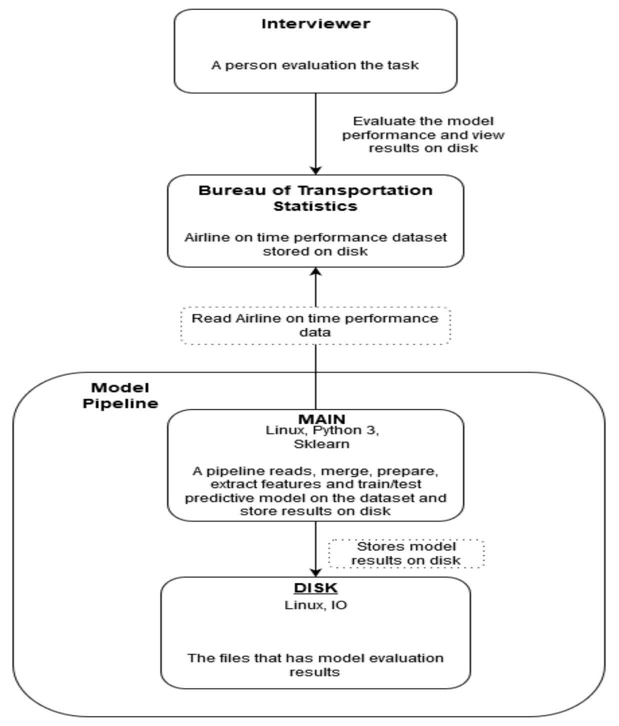
XGBoost feature Importance:



PIPELINE ARCHITECTURE

- As, I have experience that Jupyter Notebooks has more issues while model deployment for this reason I always prefer to right code in more structured way. Below is the context and container level diagrams of my model pipeline. Structured code is always easy to understand by data engineers.


Interviewer
A person evaluation the task
Evaluate the model performance and view results on disk
Bureau of Transportation Statistics
Airline on time performance dataSet stored on disk
Read Airline on time performance data
Model Pipeline
A pipeline reads, merge, prepare, extract features and train/test predictive model on the dataset and store results on disk


Interviewer
A person evaluation the task
Evaluate the model performance and view results on disk
Bureau of Transportation Statistics
Airline on time performance dataset stored on disk
Read Airline on time performance data
Model Pipeline
MAIN
Linux, Python 3, Sklearn
A pipeline reads, merge, prepare, extract features and train/test predictive model on the dataset and store results on disk
Stores model results on disk
DISK
Linux, IO
The files that has model evaluation results

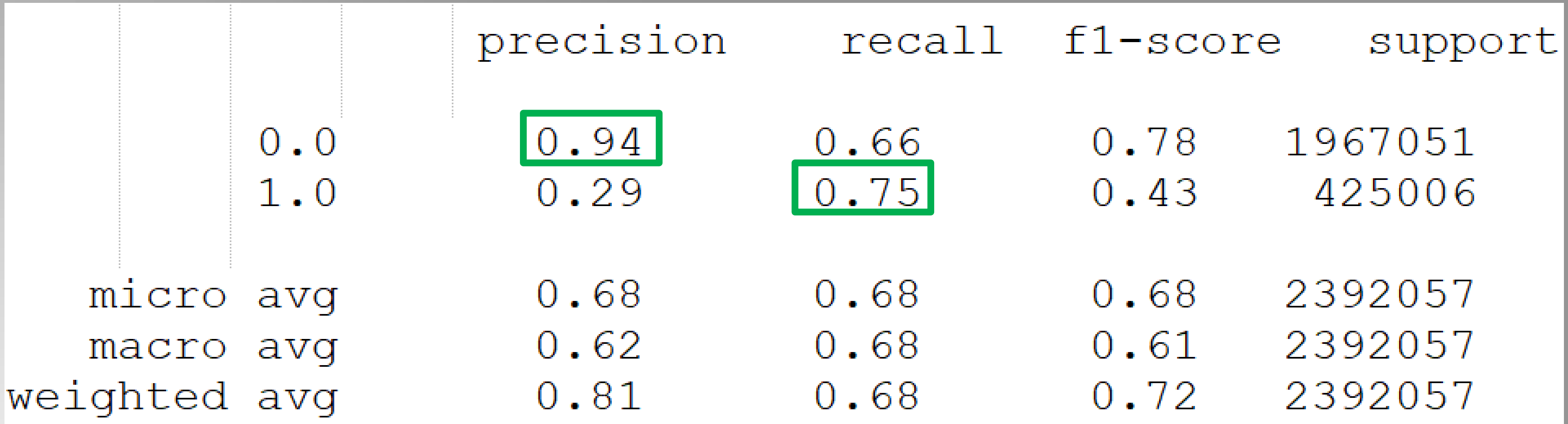
Further Improvements

Model Performance Improvement:

- I believe that we can improve the performance of the model further by incorporating different information into our data:
- These information are as follows:
  - I strongly believe that we can enhance our model performance by incorporating Weather information such as,
    Humidity, Wind speed, Fog, Storm and Temperature etc.
  - We can also improve the performance by incorporating events information to our data such as, Earthquake, price of fuel and Maintenance of the aircraft etc.
  - One other possible direction to improve model is to incorporate security information to the model for instance, Immigration, security check, and aviation safety information etc.
- For verification purpose I incorporate the weather information to see how my model will behave. For this purpose I used weather information for year 2011 and 2012 from Iowa state university. This dataset is publicly available and can be downloaded from here .

Further Improvements

XGBoost with weather information:

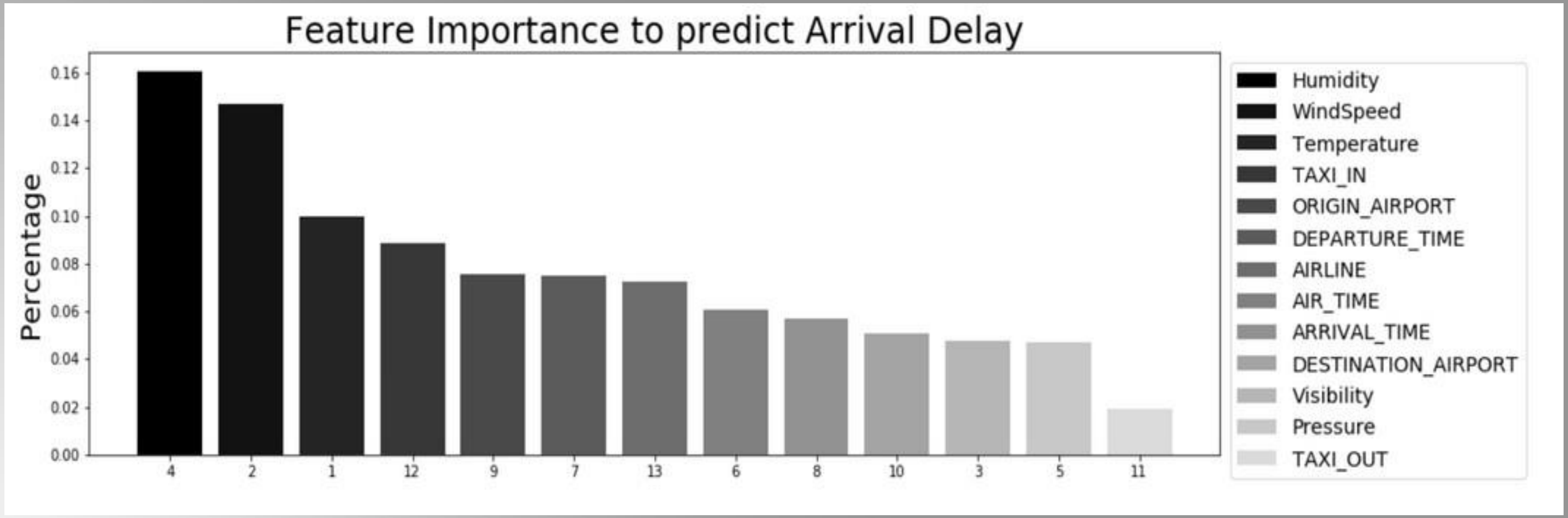


```
              precision    recall  f1-score   support

         0.0       0.94      0.66      0.78   1967051
         1.0       0.29      0.75      0.43    425006

   micro avg       0.68      0.68      0.68   2392057
   macro avg       0.62      0.68      0.61   2392057
weighted avg       0.81      0.68      0.72   2392057
```

- After incorporating weather information to our model we can see a significant lift in precision and recall. That proves our hypothesis.

Further Improvements

XGBoost feature Importance with weather information:



- We can clearly see from the above feature importance that weather information plays a vital role in predicting flight arrival delays.

Further Improvements

How flight delays could be minimized:

- There are number of factors which could help us to minimize flight delays.
- These factors are as follows:
  - By predicting runway occupancy rate before flights scheduling.
  - Parking should not be far away from the runway.
  - Busy routes could be optimized.
  - By increasing the number of flights for most frequently used routes.
  - Less importance or short flights could be schedule to nearby small airports, this could help to reduce the traffic on big and busy airports.
  - Increase the number of runways also helps in reducing flight delays.
  - As, we have more number of flight delays during evening we can divide the load by splitting some of the short flights to morning.
  - We can also minimize flight delay by increasing number of security and immigration counters for crowded flights.

FUTURE DIRECTION

Improve model performance:

- We can see from earlier slides that incorporating more information like weather etc. improves the performance of the prediction so, one of the future direction could be incorporating these information.
- One possible direction could be use of all data i.e. from year 1987 to 2012. This could be possible on AWS tenant, spark and Hadoop or online machine learning like incremental model techniques.
- Increasing in size of the hyper parameter tuning grid could also lead us to model improvement so, this could be another possible direction.

Improvements could be made in source code:

- All the scripts could be organized in proper directory formats for simplicity.
- Directory and file paths could be manage in a proper way like config file.
- We can introduce unit tests, which are need when we deploy the model.
- For back tracking I am saving files at every step like after read, prepare, feature engineering: we can avoid this but its useful to back track any problem.

Thank You.... !!!!