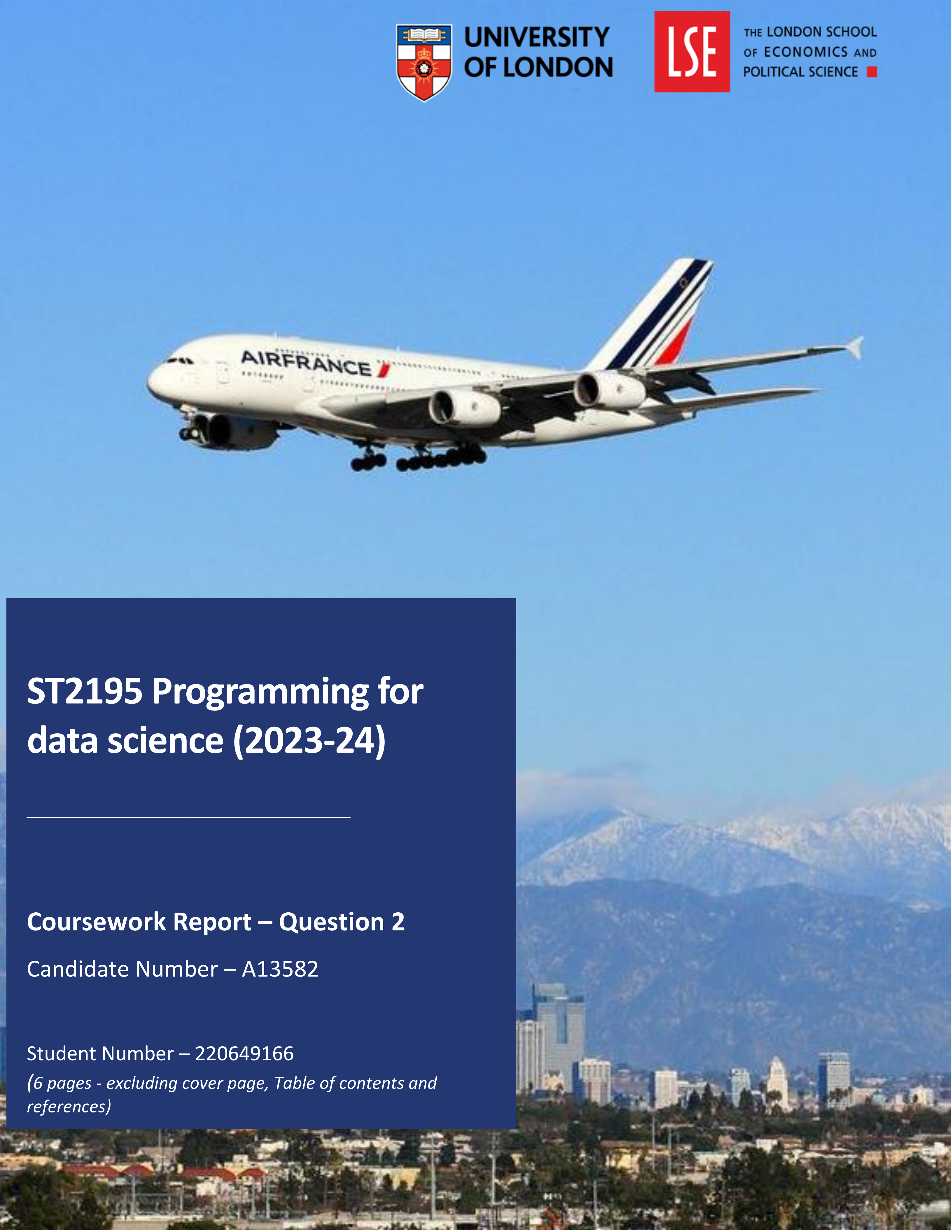
UNIVERSITY OF LONDON

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE



ST2195 Programming for data science (2023-24)

Coursework Report – Question 2

Candidate Number – A13582

Student Number – 220649166

(6 pages - excluding cover page, Table of contents and references)

Table of Contents

01. Introduction:

The 2009 ASA Statistical Computing and Graphics Data Expo provides a comprehensive dataset that includes specific details about flight arrivals and departures for all commercial flights operated by major carriers in the United States. This vast dataset, which spans from October 1987 to April 2008, includes approximately 120 million records and provides unique insights into the dynamics of air travel over a period of two decades. The datasets for the years 2006 and 2007, sourced from the Harvard website (https://doi.org/10.7910/DVN/HG7NV7), represent the most recent and extensive collections of observations available for analysis. Consequently, these datasets were selected for investigation using both R and Python programming languages. Furthermore, supplemental data files comprising information on airports and flights were used to support the study.

In this report, we will cover five core sections: Data cleaning and pre-processing procedure, investigating optimal flight times and days of the week to reduce delays, would older airplanes experience higher delays and finally developing a model that forecasts delays.

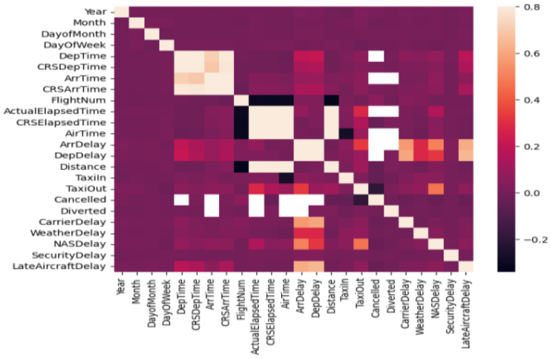
02. Data Cleaning and Pre-processing Procedure:

The datasets for the years 2006 and 2007 were merged into a single dataset due to identical column structures, while the supplementary datasets on airports and planes remained separate for analysis as required. Initial data cleaning involved removing duplicate rows across all datasets. When analyzing flight delays, values less than 0 (indicating early departure), equal to 0 (on-time departure), and greater than 0 (delayed departure) were dropped to focus on relevant data. Null values were dropped only for the columns being analyzed, avoiding unnecessary loss of data. Outliers were identified but retained to preserve valuable information on delays. For modeling purposes, indicator variables were created, and numerical data were standardized where necessary to ensure robust analysis.

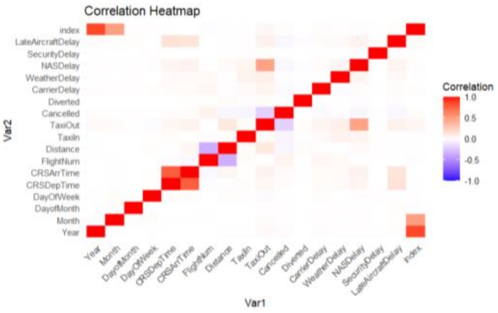
03. Investigating optimal flight times and days of the week to reduce delays:

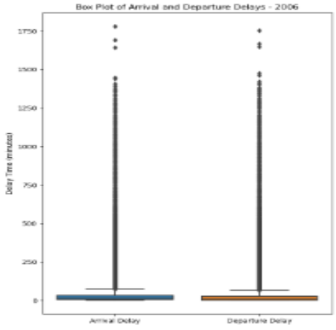
Initially, from the 2006 and 2007 merged dataset we create two separate datasets for Arrival Delay and Departure Delay, then include ‘Month’, ‘DayOfWeek’ and ‘CRSDepTime’ for each year. The analysis covers arrival and departure delays, which has a strong correlation, showing a substantial relationship in the correlation heat map below.

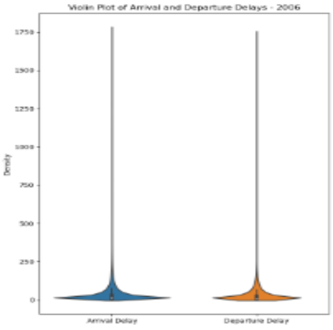
Python

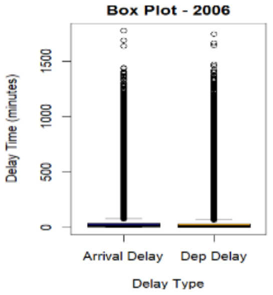


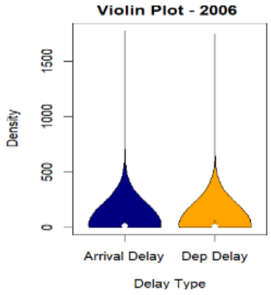
R

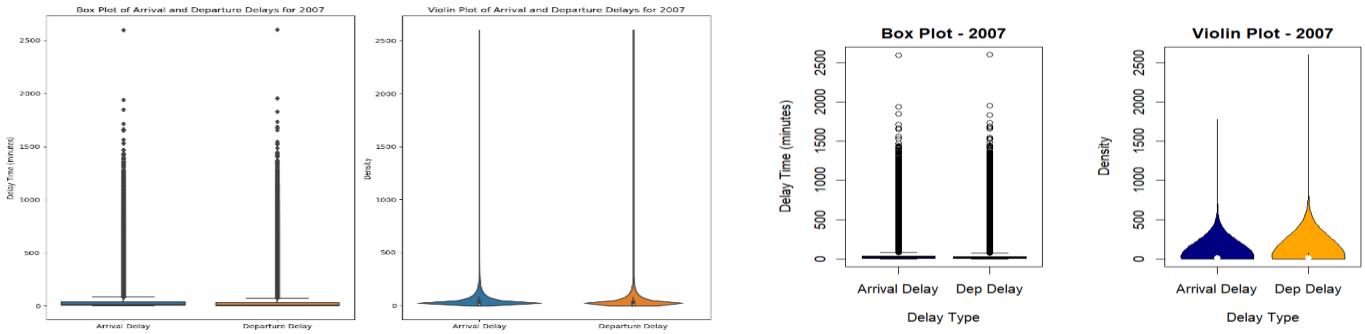












[Figure – 1]: Visualizing the distribution of Arrival and Departure delays for each year

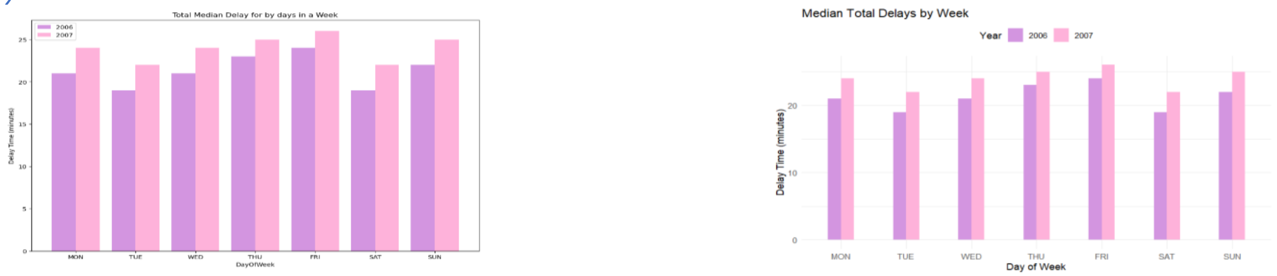
Boxplots are used to identify outliers in the distribution and remove them for better representation. The positively skewed distributions and numerous outliers in Figure 1 indicate that the median should be used for analysis because the mean is unsuitable for the data's characteristics.

Line graphs and bar charts are created to analyze the optimal times for minimizing delays, showcasing median arrival and departure delays over a 24-hour and seven-day period. To enhance clarity, the total median delay is calculated by summing the median arrival and departure delays, representing the overall influence on delays.

3.1 Best day of the week to reduce delays:

Python

R



[Figure – 2]: Best Day of the week to reduce delays in each year

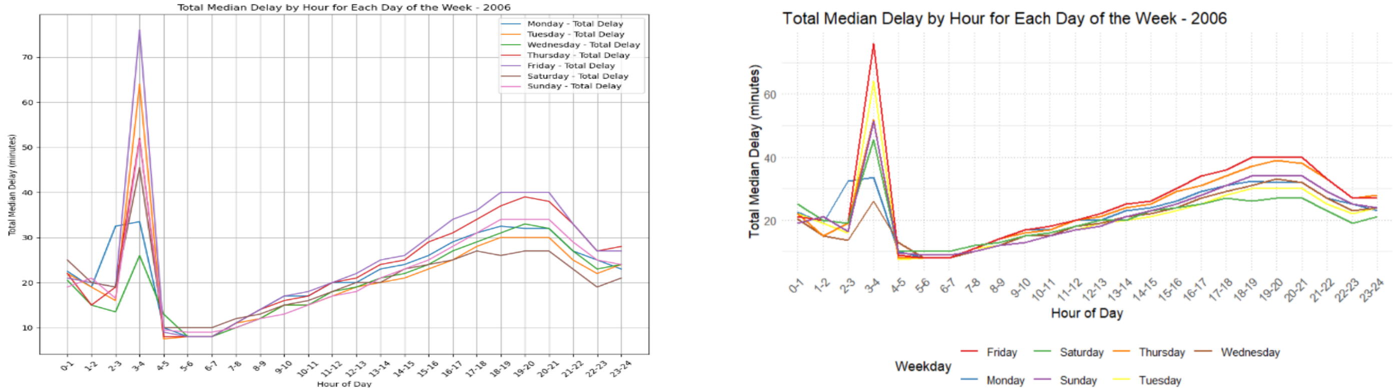
Figure 2, shows that Tuesday and Saturday are the most favorable days of the week when flying due to its consistently lowest total median delay in both years.

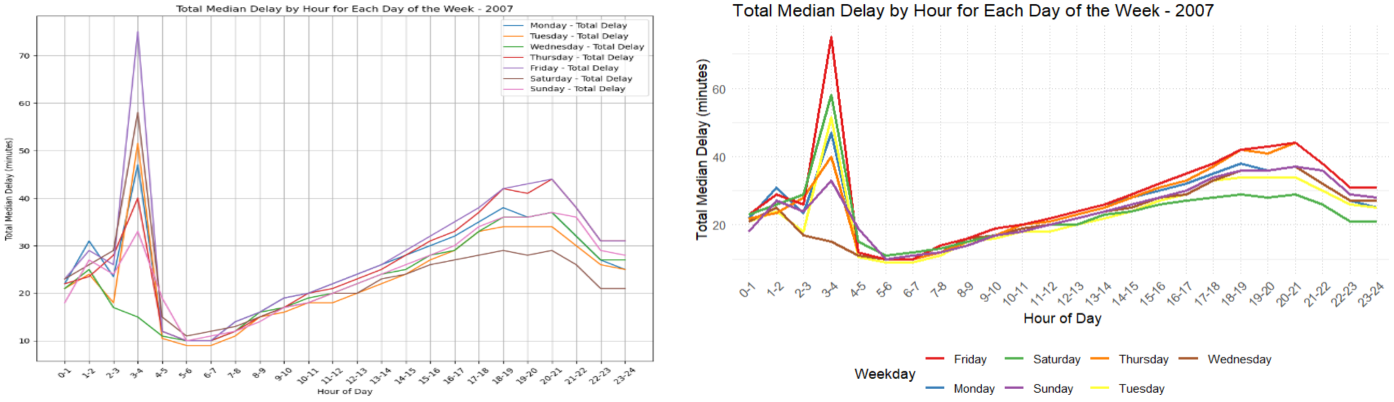
3.2 Best Times and days of the week to fly to minimize delays:

Using Figure 3 below, we can say that 4am - 7am are the best hours for all seven days of the week in both years considering the lowest total median delays. Further narrowing the best times to travel, 4am to 6am on Tuesdays of 2006 and 5am to 7am Tuesday of 2007 found to offer the lowest median delays. To minimize delays during the peak periods indicated on the graph for 3-4 am and 6-9 pm, it's advisable to travel on Wednesdays and Saturdays in both 2006 and 2007, as they consistently exhibit the lowest median delays.

Python

R





[Figure – 3]: Best Times and Days of the week to minimize delays

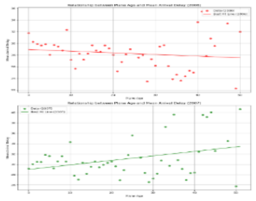
04. Evaluating “whether Older planes suffer more Delays”:

The evaluation involves studying the relationship between plane age and delays by combining the plane-dataset based on tail number and calculating the plane age by taking the difference between the columns ‘Year’ and ‘YearOfManufacture’ (which is the ‘year’ column from plane-dataset).

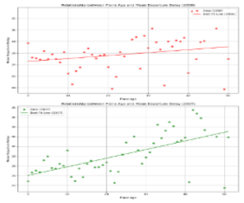
4.1. Using only Arrival and Departure Delays:

Utilizing two separate cleaned datasets for Arrival Delay and Departure Delay, each containing columns 'TailNum', 'Year', 'YearOfManufacture', and 'Planeage', we calculate the mean arrival and departure delays for each plane age. Subsequently, we determine the correlation coefficient between plane age and delays.

Python

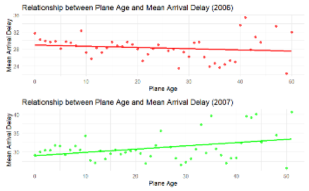


```
Correlation between Plane Age and Mean Arrival Delay (2006): -0.14766160252866734
Correlation between Plane Age and Mean Arrival Delay (2007): 0.3415208513265151
```

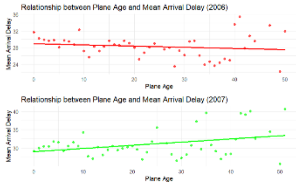


```
Correlation between Plane Age and Mean Departure Delay (2006): 0.26142912910859045
Correlation between Plane Age and Mean Departure Delay (2007): 0.6289047086329791
```

R



```
Correlation between Plane Age and Mean Arrival Delay (2006):  -0.1476616
Correlation between Plane Age and Mean Arrival Delay (2007):  0.3415209
```



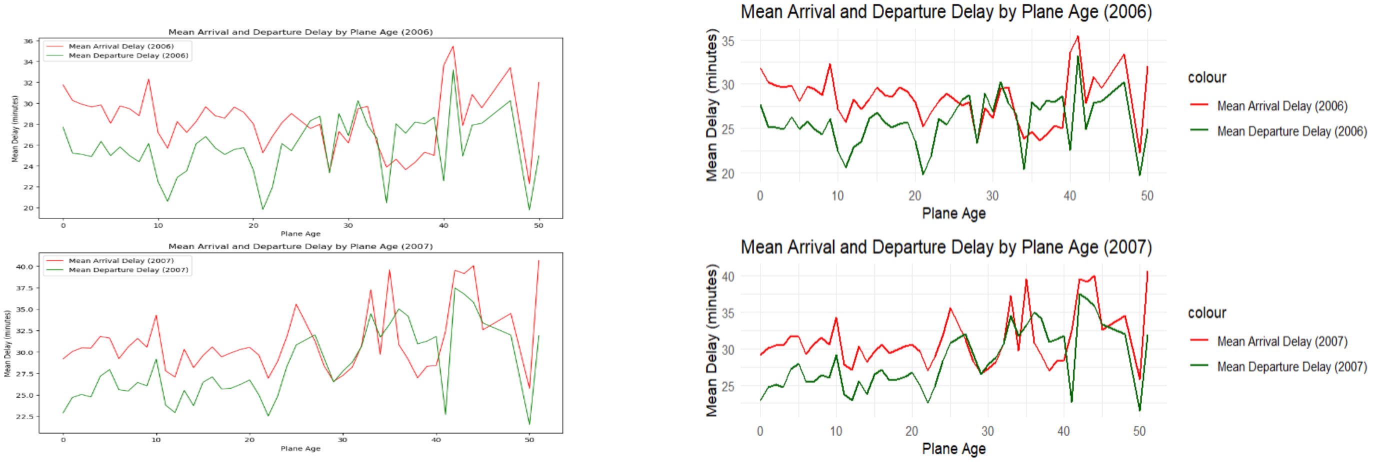
```
Correlation between Plane Age and Mean Departure Delay (2006):  0.2614291
Correlation between Plane Age and Mean Departure Delay (2007):  0.6289047
```

[Figure – 4]: Correlation between Planeage and Mean Arrival and Departure Delays

Based on the figure above, we observe that the mean arrival delay for 2006 exhibits a negative correlation. Additionally, the mean arrival delay of 2006 and the mean departure delay of 2007 show a weak positive correlation, except for the 2007 mean departure delay, which demonstrates a strong correlation.

The line plot in Figure 5, reveals fluctuations in delay as plane age increases, but it does not suggest a clear trend, whether increasing or decreasing. There is a significant increase in mean delay around 10 years of age in 2006 and between 20-30 years in 2007, followed by sharp drops at 50 years in 2006 and at 40 and 50 years in 2007, where the lowest mean delays are observed. Thus, the plot does not conclusively demonstrate that older planes experience more delays.

Python *R*



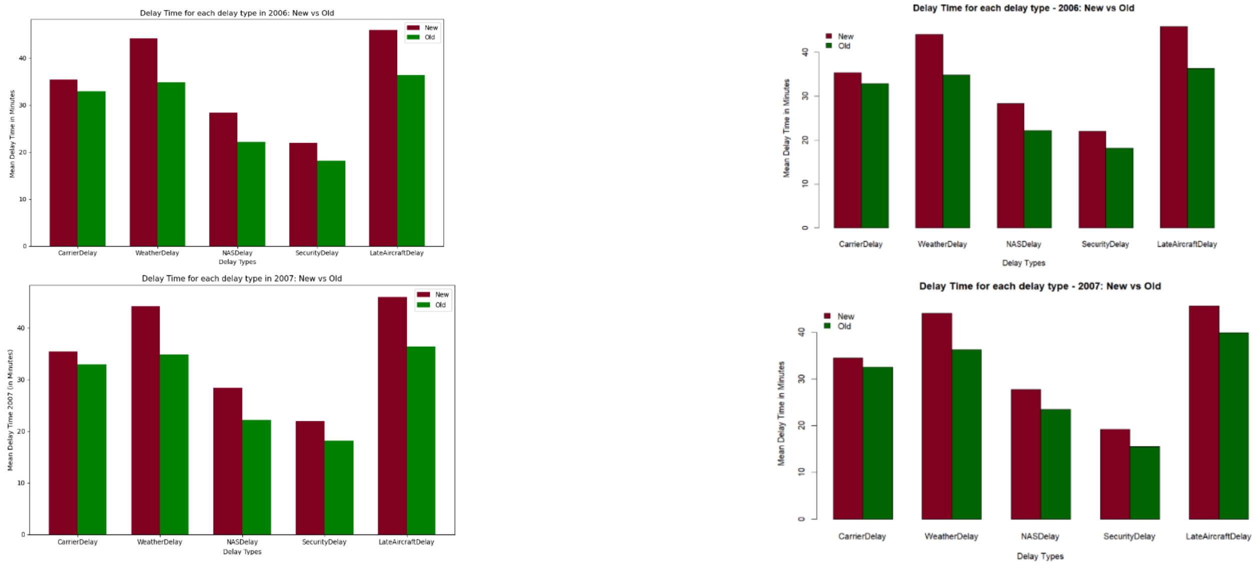
[Figure – 5]: Mean Arrival and Departure Delays by plane age

The line plot in Figure 5, reveals fluctuations in delay as plane age increases, but it does not suggest a clear trend, whether increasing or decreasing. There is a significant increase in mean delay around 10 years of age in 2006 and between 20-30 years in 2007, followed by sharp drops at 50 years in 2006 and at 40 and 50 years in 2007, where the lowest mean delays are observed. Thus, the plot does not conclusively demonstrate that older planes experience more delays.

4.2. Using the Five Delay types:

We segmented the cleaned datasets containing five delay types into five separate datasets based on their delay type. Rows with a delay type of 0 were removed to ensure accurate mean calculations. The same columns used before for arrival and departure delays were retained without ‘ArrDelay’ and ‘DepDelay’ columns. Additionally, to simplify the analysis, we introduced a new column called "plane age condition" based on plane age. Considering that “most airplanes are dismantled when they reach 18 years old” an assumption is made to categorize planes as "new" if they are less than or equal to 20 years old and "old" otherwise, in this column.

Python *R*



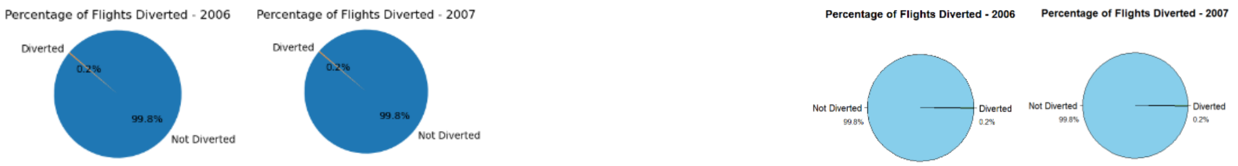
[Figure – 6]: Mean Delay Types by plane age condition for 2006 and 2007 years

Figure 6 illustrates that mean delays are consistently higher for new planes compared to old planes, which provides clear evidence indicating that older planes don't always experience more delays.

05. Interpreting the ‘Logistic Regression Model’ for the probability of diverted US flights for each year:

The analysis started by combining the 2006 flight dataset with airports and carriers’ datasets. Upon initial assessment, we noticed a significant class imbalance, with far fewer diverted flights (class 1) compared to non-diverted flights (class 0). To tackle this imbalance, we focused on cleaning the data by removing null values. We chose to remove certain columns ('city_origin', 'state_origin', 'city_dest', 'ActualElapsedTime') that had null values instead of getting rid of all null rows, which could unfairly affect the minority class (class 1). After that, we picked out the most important numerical features for the model where we removed that ‘TaxIn’ due to its unusually high selection score for year 2007. Then, we dealt with categorical data and split the dataset into parts for training and testing. Then all these steps above were done on 2007 dataset also. Before moving on to the model, we made sure to check the imbalance between classes for both years.

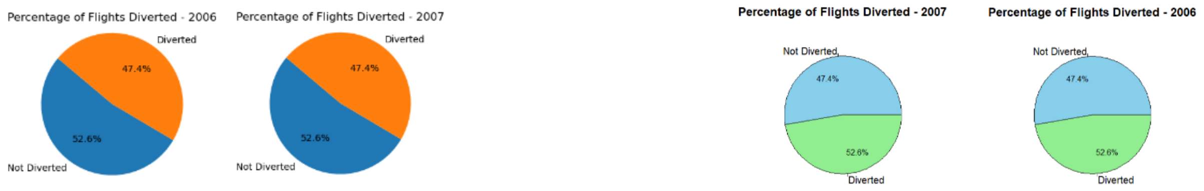
Python *R*



[Figure – 7]: Diverted Counts before balancing the dataset

To balance the ‘Diverted’ classes, we import the resample library and utilize the resampling functions to oversample class 0 and under-sample class 1, assigning desired counts (180,000 and 162,000 respectively). These desired counts ensure that there is a percentage difference between the class counts below 20%, as given below in Figure 7.

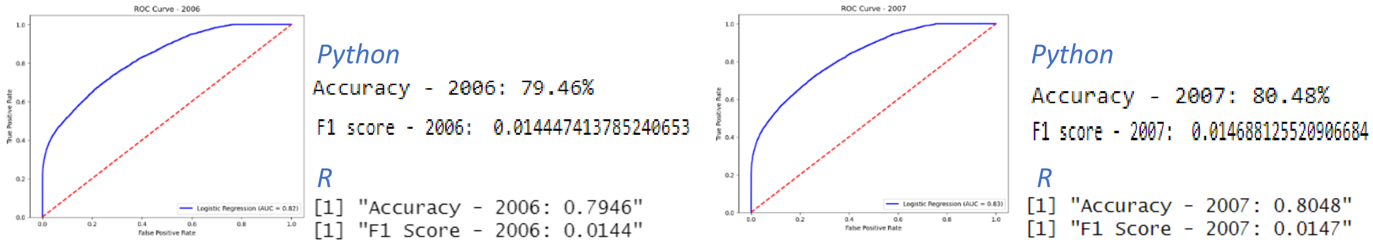
Python *R*



[Figure – 8]: Diverted Counts after balancing the dataset

5.1. ROC Curves:

The overall accuracy achieved is around 79% and 80%, indicating that the model correctly predicts approximately 79% and 80% of all instances for 2006 and 2007 respectively as shown below.



Python

```
Accuracy - 2006: 79.46%
F1 score - 2006:  0.014447413785240653
```

R

```
[1] "Accuracy - 2006: 0.7946"
[1] "F1 Score - 2006: 0.0144"
```

Python

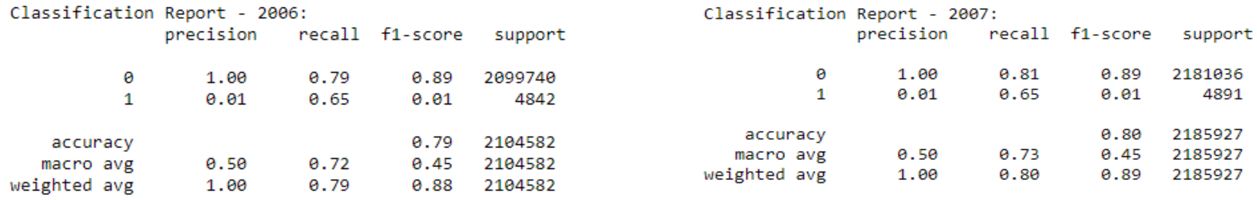
```
Accuracy - 2007: 80.48%
F1 score - 2007:  0.014688125520906684
```

R

```
[1] "Accuracy - 2007: 0.8048"
[1] "F1 Score - 2007: 0.0147"
```

[Figure – 9]: Logistic Regression Model – ROC Curve

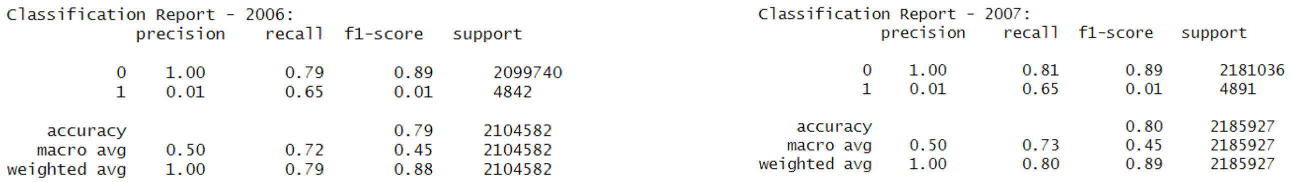
Python



```
Classification Report - 2006:
              precision    recall  f1-score   support

           0       1.00      0.79      0.89   2099740
           1       0.01      0.65      0.01      4842

    accuracy                           0.79   2104582
   macro avg       0.50      0.72      0.45   2104582
weighted avg       1.00      0.79      0.88   2104582
```

```
Classification Report - 2007:
              precision    recall  f1-score   support

           0       1.00      0.81      0.89   2181036
           1       0.01      0.65      0.01      4891

    accuracy                           0.80   2185927
   macro avg       0.50      0.73      0.45   2185927
weighted avg       1.00      0.80      0.89   2185927
```

R



```
Classification Report - 2006:
              precision    recall  f1-score   support

           0       1.00      0.79      0.89   2099740
           1       0.01      0.65      0.01      4842

    accuracy                           0.79   2104582
   macro avg       0.50      0.72      0.45   2104582
weighted avg       1.00      0.79      0.88   2104582
```

```
Classification Report - 2007:
              precision    recall  f1-score   support

           0       1.00      0.81      0.89   2181036
           1       0.01      0.65      0.01      4891

    accuracy                           0.80   2185927
   macro avg       0.50      0.73      0.45   2185927
weighted avg       1.00      0.80      0.89   2185927
```

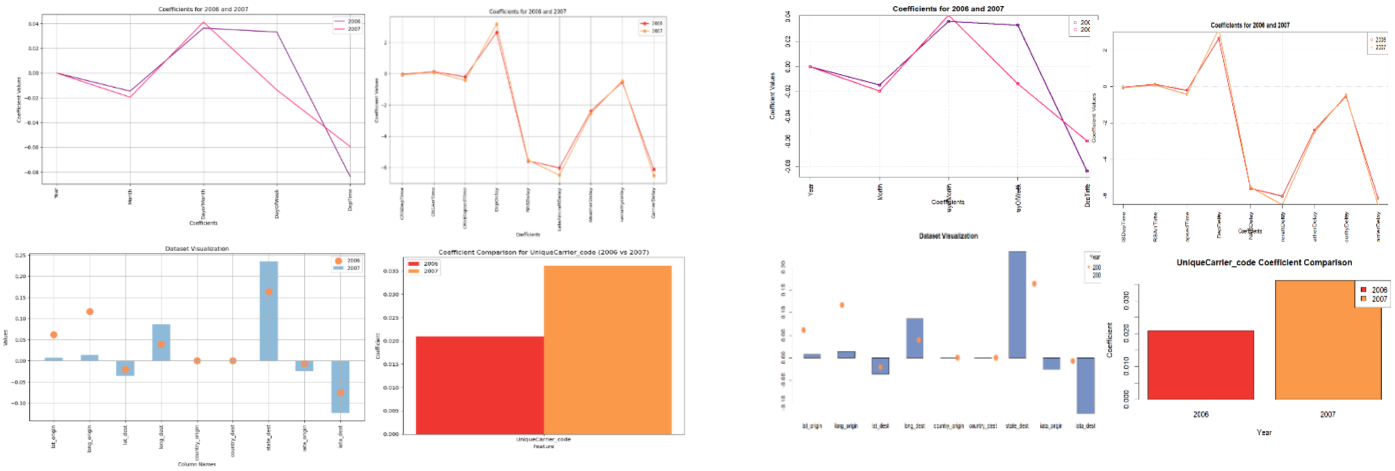
[Figure – 10]: Classification Report

The poor performance, characterized by low precision and recall for class 1 despite high accuracy for class 0, is a consequence of the significant class imbalance. With the model favoring the majority class, it struggles to effectively predict instances from the minority class. To overcome this issue, we can try to employ models that are more resilient to unbalanced datasets, such as “Random Forest, Gradient Boosting Machines, or techniques like XGBoost or LightGBM”. Furthermore, altering class weights can aid in the model's capacity to correctly categorize both classes.

5.2. Visualizing the coefficients across two years:

The coefficients extracted from the previously fitted logistic regression models were merged to form datasets for two consecutive years, with features as columns and corresponding coefficient values. Subsequently, separate datasets were created by segregating features based on the four attributes queried: departure date, scheduled departure and arrival times, coordinates and distance between departure and planned arrival airports, and carrier. The datasets were visualized using ordered plots, arranged from top-left to right, then bottom-left to right for clarity, as illustrated in Figure 5.

Python *R*



[Figure – 11]: Visualizing the Coefficients across two years

The coefficients from the logistic regression model indicate the relationship between the features and the probability of diverted US flights. A positive coefficient implies that a rise in the feature's value corresponds with a greater chance of diversion, whereas a negative coefficient indicates the contrary. In the first plot ‘DayOfWeek’ has a positive coefficient in 2006 and a negative coefficient in 2007 which indicates that flights scheduled on certain days of the week for 2006 are more likely to be diverted than in 2007. Flight diversion patterns remained consistent between 2006 and 2007 for the second plot which include the features for the scheduled arrival and departure time attribute. Zero coefficients in plots 2 and 3 indicate that, according to the logistic regression model, there is no apparent relationship between flights scheduled based on those features and the likelihood of flight diversions. In plot 4, both coefficients imply a positive association between the carrier represented by the code and the likelihood of flight diversions in both 2006 and 2007. Further analysis is necessary to understand the specific reasons behind these relationships and its implications.

06. References:

- Volpi, Gonzalo Ferreiro (2019). "Class Imbalance: a classification headache". Available at: https://towardsdatascience.com/class-imbalance-a-classification-headache-1939297ff4a4

- Brownlee, Jason (2020). "A Gentle Introduction to Imbalanced Classification". Available at: https://machinelearningmastery.com/what-is-imbalanced-classification/

- Yang Feng and Jianan Zhu (2022). "R Programming: Zero to Pro"- (GitHub). Available at:
  Main link: https://r02pro.github.io/
  Other sub-links for the r codes taken:
  https://r02pro.github.io/violin.html, https://r02pro.github.io/select-variables.html, https://r02pro.github.io/bar-charts.html, https://r02pro.github.io/bar-charts.html, https://r02pro.github.io/summary-geom.html, https://r02pro.github.io/import-data.html, https://r02pro.github.io/import-data.html#handling-missing-values etc.

- Pierre L, (2016) modified 3 years ago. "R replicate sample function without replacement". Available at: https://stackoverflow.com/questions/37422370/r-replicate-sample-function-without-replacement

- Schweinberger, Martin (2021). "Data Visualization with R". Available at: https://slcladal.netlify.app/dviz.html

- Geeksforgeeks, updated in 2023. "Data Visualization in jupyter notebook". Available at: https://www.geeksforgeeks.org/data-visualization-in-jupyter-notebook/
  "Article Tags: AI-ML-DS With Python, Geeks Premier League 2023, Jupyter-notebook, AI-ML-DS, Data Visualization, Geeks Premier League"

END.