UNIVERSITY OF LONDON

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

ST2195 – Programming for Data Science (Coursework Report)

Module – ST2195

Module Title - Programming for data science

UOL Student Number – 220639927

TABLE OF CONTENTS

INTRODUCTION 2

PART 1 2

PART 2 2

PART 1 3

PART 2 4

DATA CLEANING PROCESS 4

(a) 4

(b) 5

(c) 7

INTRODUCTION

PART 1

The first part of the report delves into the application of the Metropolis-Hastings algorithm of the Markov Chain Monte Carlo algorithm, where random samples are generated for the probability distribution;

$$f(x) = \frac{1}{2}\exp(-|x|)$$

Part A involves generating samples, constructing a histogram, KDE plot, and plotting the f(x) on the same figure. The sample mean and the sample standard deviation are calculated as well.

Subsequently, the convergence of the algorithm was assessed using the $\hat{R}$ value. The variation of the $\hat{R}$ value was evaluated by plotting $\hat{R}$ values over a range of 's' values.

PART 2

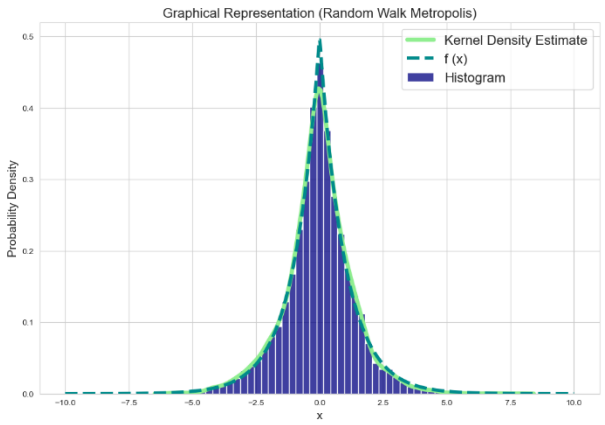
The second part conducts a critical analysis of flight data of all commercial flights on major carriers within the USA from October 1987 to April 2008. The datasets from 2006 and 2007 were used to conduct the analysis. After a thorough data cleaning process, the optimal times and days to minimize delays for each year, as well as the impact of aircraft age on delays on a year-to-year basis were examined. Followed by a separate data cleaning process, a logistic regression model to predict the probability of flight diversions was developed.

Note: The solutions for Part 1 were obtained, and the necessary data analysis for Part 2 was conducted using Python and R, generating identical conclusions and illustrations.

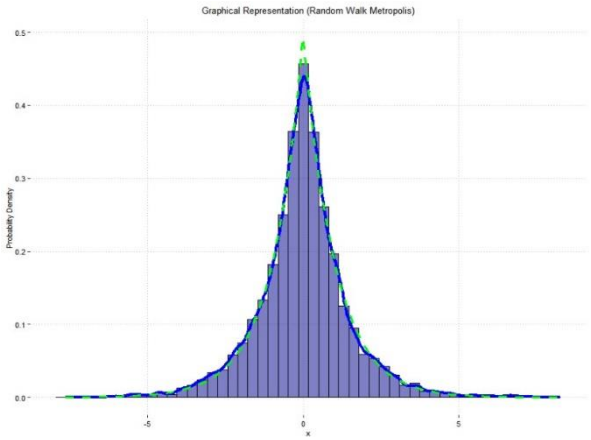
PART 1

Part A

The probability density function f(x) is defined and initial values x0 = 0, N = 10000, and s = 1 are assigned. A loop iterating N times is created, generating new samples. The proposed sample x* is generated from a normal distribution with the mean as the last generated sample and standard deviation s. The acceptance ratio (r) is calculated and a random number u from the uniform distribution between 0 and 1 is generated. If log u < log r, x* is accepted; otherwise, the last generated sample is kept. **The Monte-Carlo estimates of the sample mean and standard deviation were 0.01369 and 1.37329 respectively**. The generated samples are graphically represented below using a histogram and KDE plot, overlaying f(x).



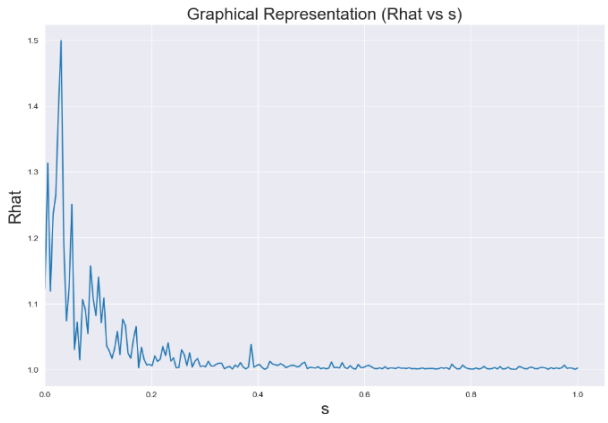
Python



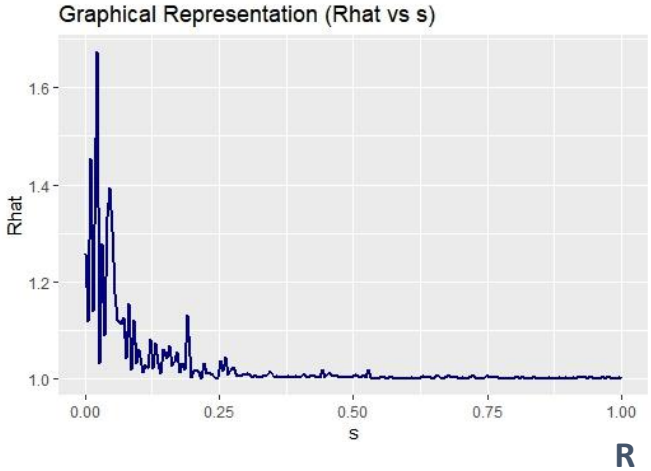
R

Part B

The procedure to assess convergence using $\hat{R}$ includes generating multiple chains of samples, obtaining the sample mean (Mj) and sample variance (Vj) for each chain, and then the overall within sample variance (W), followed by the overall sample mean(M) and between sample variance (B). The $\hat{R}$ value obtained was **1.1557654972974798.** $\hat{R}$ is graphically represented against a range of s values, **indicating convergence.**



Python



R

PART 2

DATA CLEANING PROCESS

The data cleaning process for flight analysis was conducted twice; once for parts (a) and (b) and again for part (c).

The initial data cleaning process involved combining the datasets from 2006 and 2007, subsequently removing null values and rectifying outliers and unreasonable values. The 'Cancellation Code' column was removed since it had 98% null values and wasn't necessary for the analysis. Subsequently, the rows containing null values were removed (If this operation was conducted without removing the column 'Cancellation Code', there wouldn't have been sufficient data to conduct the analysis). The columns 'DepTime' and 'ArrTime' contained maximum values of '2930' and '2955' that made no sense as they exceeded the 24 hours. These values were rectified to 2359. The negative values in the columns 'DepDelay' and 'ArrDelay' were assumed to be early departures and arrivals.

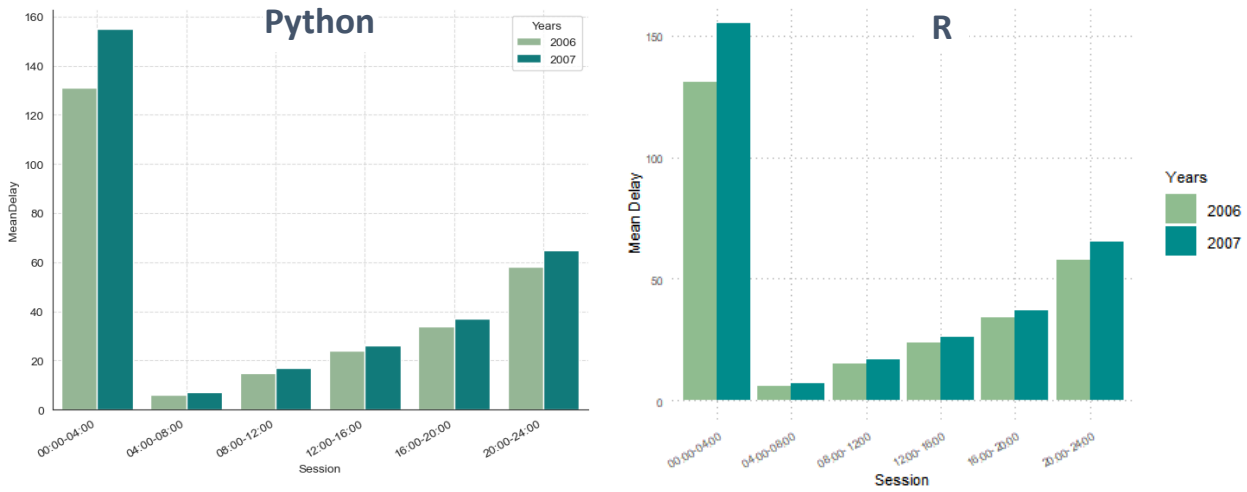
The data cleaning process for part (c) involved a minor change: instead of removing rows with null values, all columns containing null values were removed. This was necessary since all the 1 values in the 'Diverted' column were lost during the initial cleaning, making it impossible to answer part (c). Additionally, the maximum value in 'CRSDepTime' was rectified from 2400 to 2359.

(a) What are the best times and days of the week to minimize delays each year?

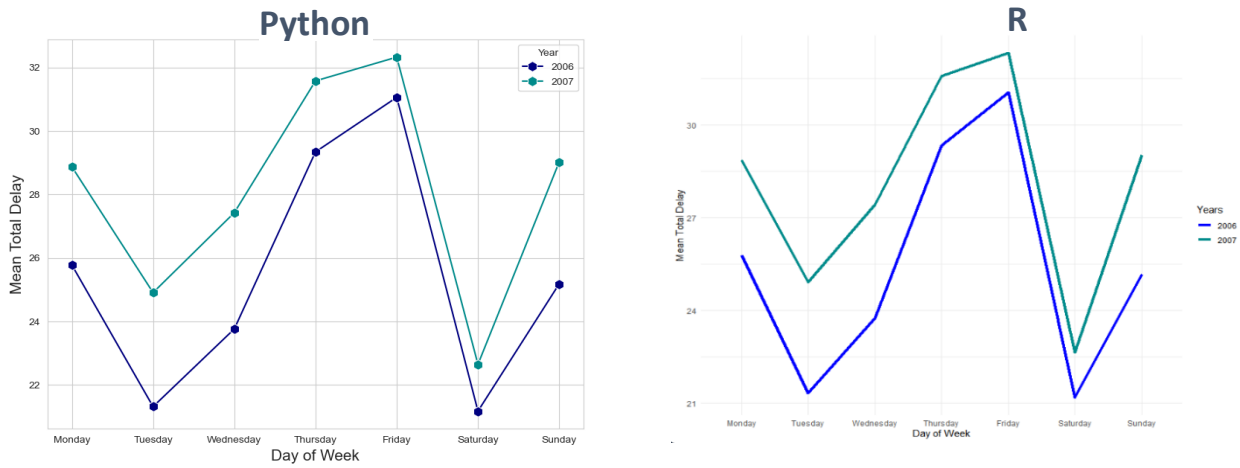
A new column 'TotalDelay' was created by adding the columns 'DepDelay' and 'ArrDelay'. All the negative values in the column 'TotalDelay' were set to 0, since the analysis focuses on delays rather than early flights. This procedure was followed for part (b) as well.

To analyze **the best times to minimize delays each year**, the 24 hours were partitioned into 6 sessions, and a new column 'Session' was added to the data frame. The data frame was split into the two years, analyzing both years separately. **The best session of the day to minimize delays was 04.00 am to 08.00 am for both years.** The timeframe 12.00 am – 04.00 am had a significantly high mean delay for both years. A pattern was observed where the mean delay gradually increased as the day progressed.

The mean delays of the sessions for both years are graphically represented below:



To find **the best days of the week to minimize delays each year**, the mean total delay for each day of the week was obtained. The numerical values in the 'DayOfWeek' column were replaced by the corresponding days of the week (Assumption: week starts from Sunday). The analysis showed that **Saturday has the lowest mean total delay for both years**. It is observed that **Saturday, Tuesday, and Wednesday are the best days of the week to minimize delays** for 2006 and 2007.

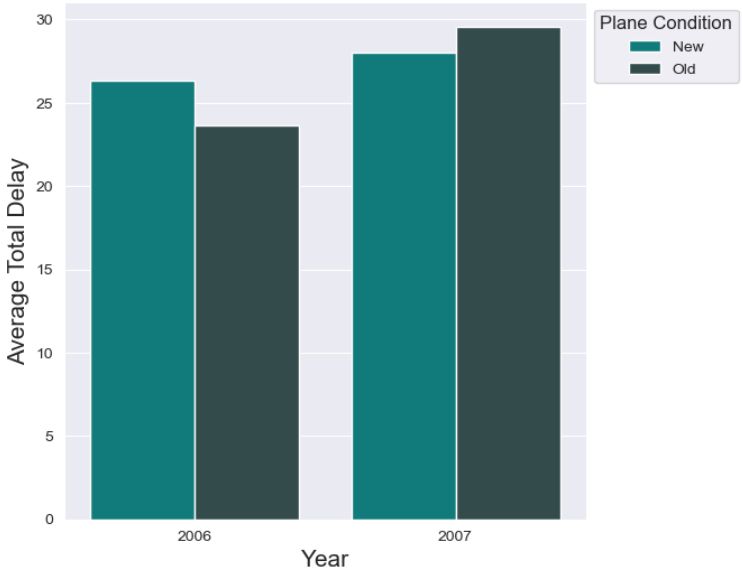


(b) Evaluate whether older planes suffer more delays on a year-to-year basis.

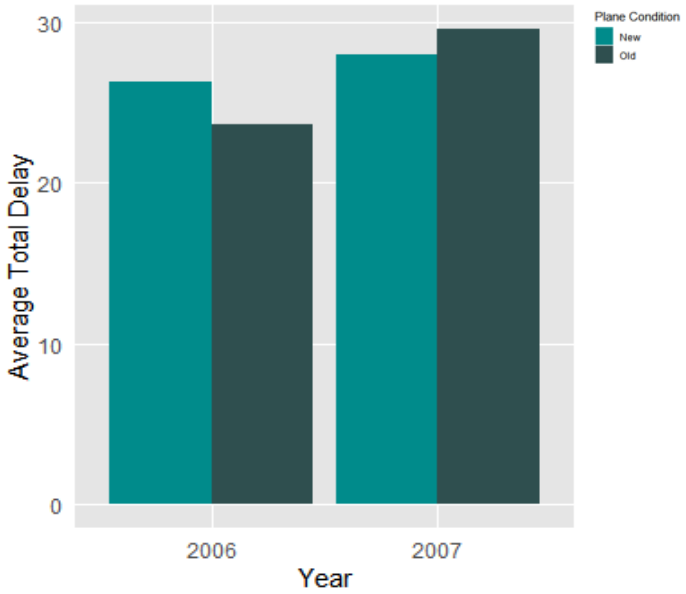
The cleaned data frame was merged with plane data on 'TailNum'. Null/unreasonable values were omitted and the entries '0000' and 'None' in the 'YearOfManufacture' column were removed. Planes were categorized as 'Old' and 'New' based on the manufactured year. **Planes manufactured before 1982 were considered 'Old'**. 'PlaneAge' was created by subtracting 'Year' from 'YearOfManufacture'. The dataset was split based on the respective years for the analysis.

Based on the analysis**, 'New' airplanes had a higher average total delay in 2006 while 'Old' airplanes had a higher average total delay in 2007**.

Python



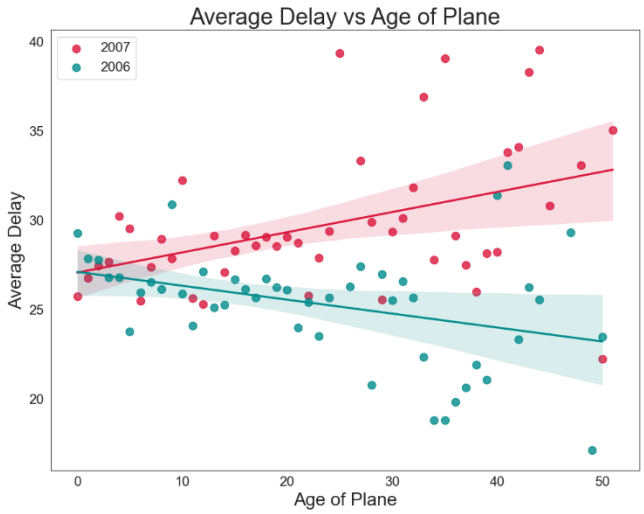
R



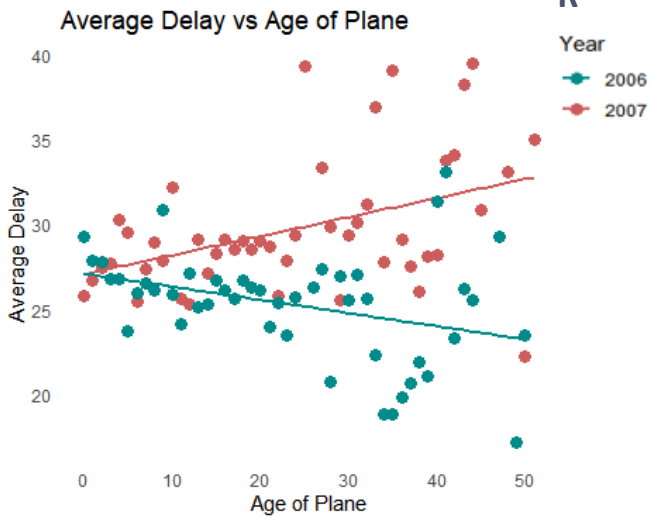
An increase in the average delay with plane age was evident in 2007, while it was intriguing that a decrease was observed in 2006. As demonstrated in the regression plots, it can be observed that older airplanes had a higher average delay in 2007 while it wasn't necessarily the case in 2006, warranting further investigation. The nature of the correlation coefficient should also be taken into consideration when arriving at conclusions.

- **Correlation coefficient (2006): -0.35** (Weak negative) **| Correlation coefficient (2007): 0.42** (Moderate positive)

Python

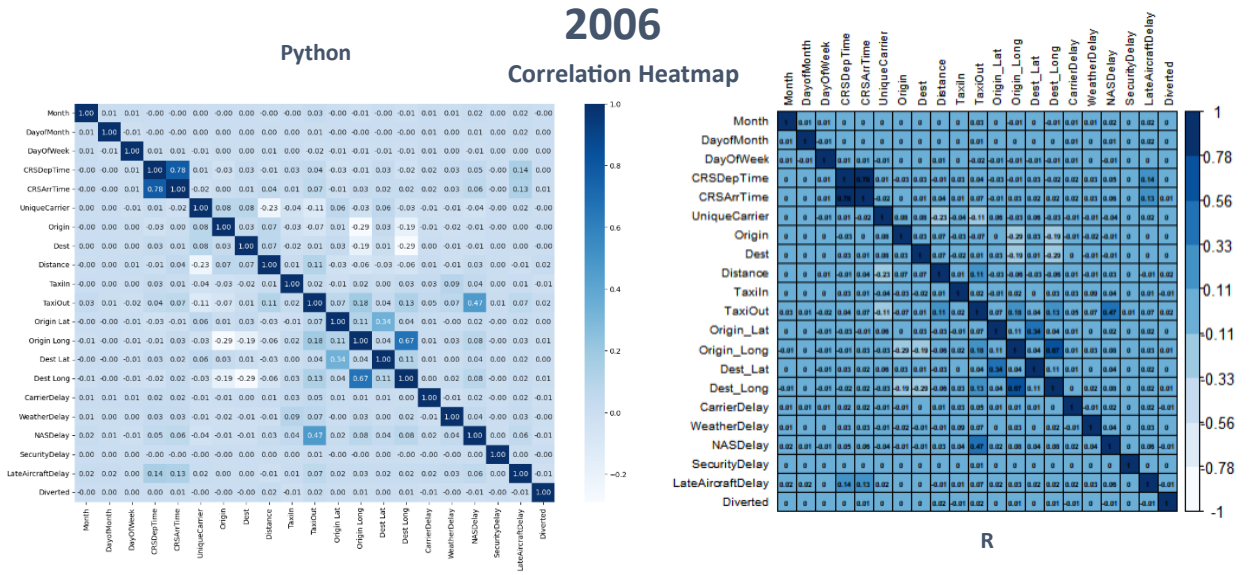


R

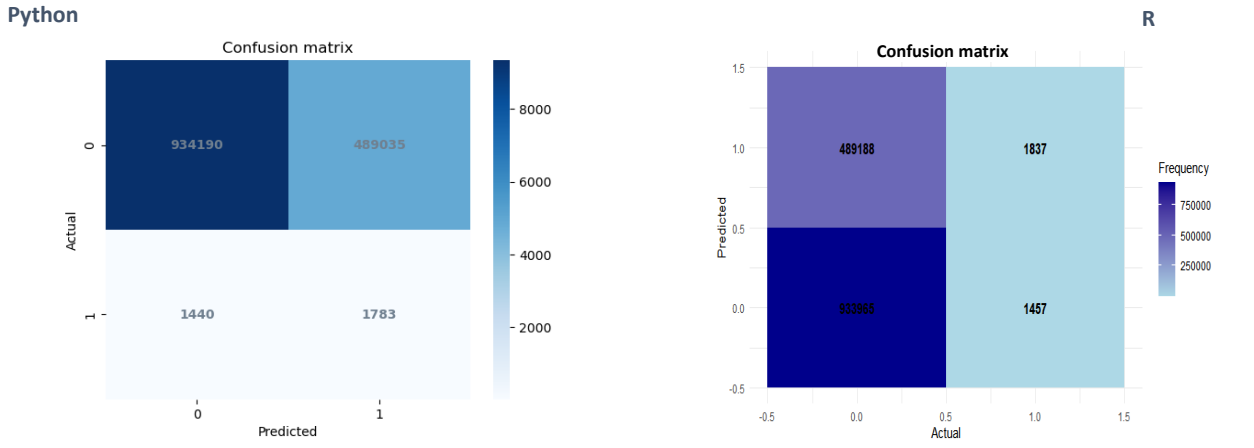


(c) For each year, fit a logistic regression model for the probability of diverted US flights.

The separately cleaned dataset, along with the 'carriers' and 'airports' datasets were used to construct the logistic regression model. The datasets were merged and the required features were obtained into one data frame. The categorical features were encoded into numerical labels using 'Label Encoder'. The value counts for 'Diverted' flights were obtained, revealing a major sample discrepancy (0 – 7125736, 1 – 33365). The dataset was split into the two years, and a correlation heatmap was created to identify any relationships among the potential features to be taken and the target variable.



Based on the correlation observed, the features taken into consideration for the logistic regression model were ''Month', 'DayofMonth', 'DayOfWeek', 'CRSDepTime', 'CRSArrTime', 'Unique Carrier', 'Distance', 'TaxiIn', 'TaxiOut', 'Origin Lat', 'Origin Long', 'Dest Lat', and 'Dest Long'. The data was split into testing and training sets and the features were standardized using the standard scaler. The majority class (0 values) in the training data were down-sampled using 'RandomUnderSampler' to match the 1 values and attempt to neutralize the severe data imbalance. The logistic regression model was developed and the confusion matrix was obtained. A classification report was obtained to showcase the accuracy of the model. This process was repeated for each year and the coefficients were visualized accordingly.




```
Accuracy: 0.6561564108891457
```

```
Accuracy: 0.656036992611713
```

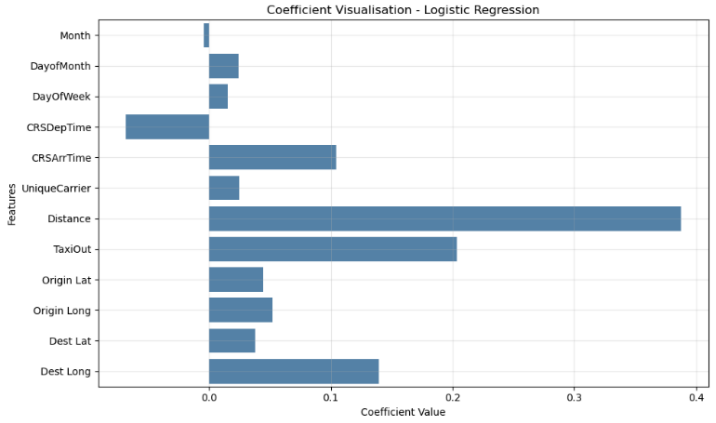
Classification Report - Python

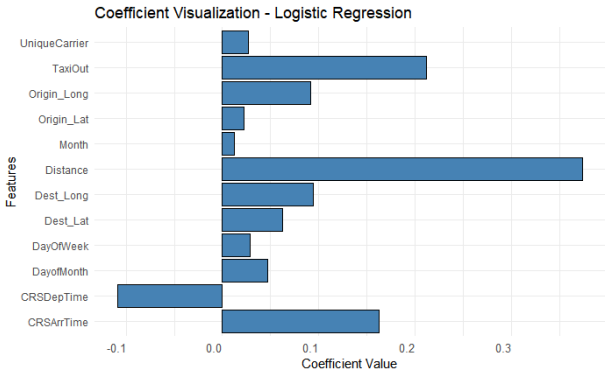
	precision	recall	f1-score	support
Not Diverted	1.00	0.66	0.79	1423225
Diverted	0.00	0.55	0.01	3223
accuracy			0.66	1426448
macro avg	0.50	0.60	0.40	1426448
weighted avg	1.00	0.66	0.79	1426448

Classification Report - R

```
              Accuracy : 0.6624
           Sensitivity : 0.662673
           Specificity : 0.555728
        Pos Pred Value : 0.998481
        Neg Pred Value : 0.003725
             Prevalence : 0.997736
         Detection Rate : 0.661172
   Detection Prevalence : 0.662178
      Balanced Accuracy : 0.609200
```

Several strategies such as under-sampling the majority class, oversampling the minority class, SMOTE (Synthetic Minority Over-sampling Technique), and Random Under Sampler were employed in an attempt to neutralize and rectify the severe class imbalance (0 – 7125736, 1 – 33365) for Diverted flights. The discrepancy continued to be a persistent challenge despite these efforts, directly contributing to the low f1-score for diverted flights, and distorting the performance of the logistic regression model, as evident in the classification report.

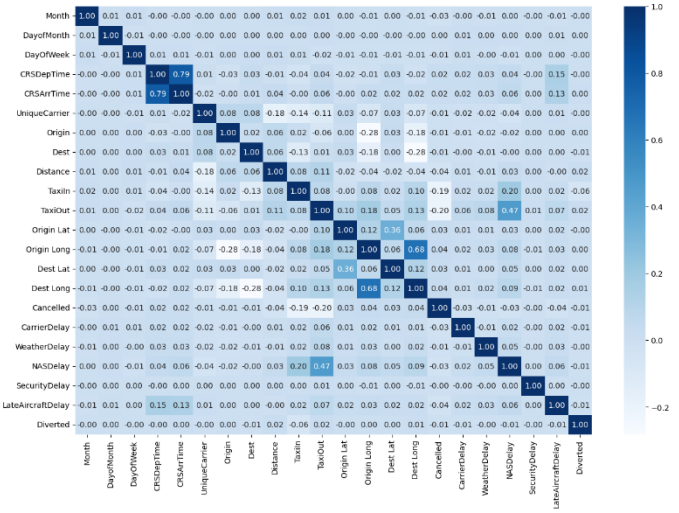




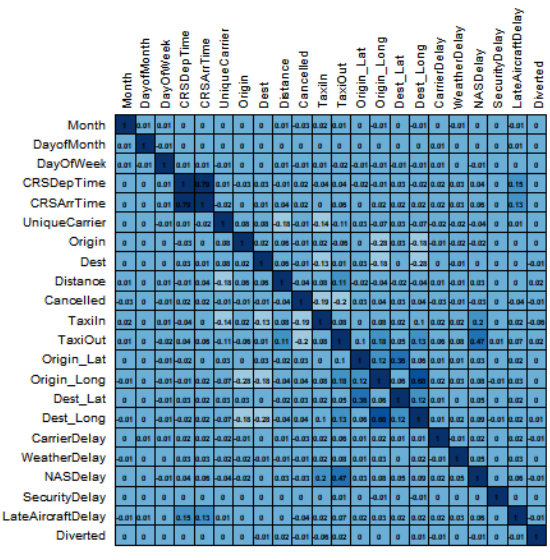
Python **R**

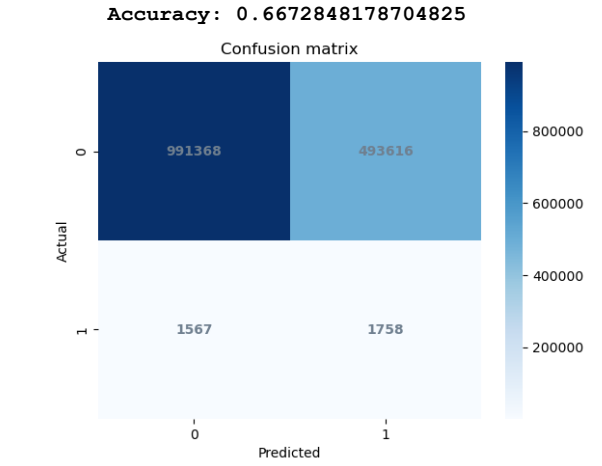
2007

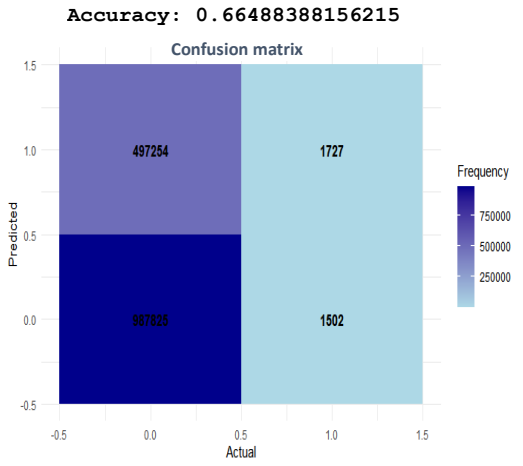
Correlation Heatmap - Python



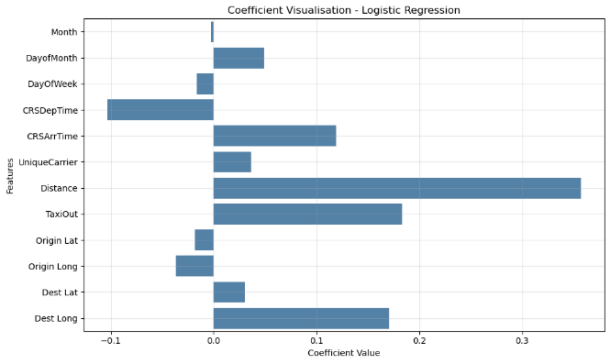
Correlation Heatmap - R



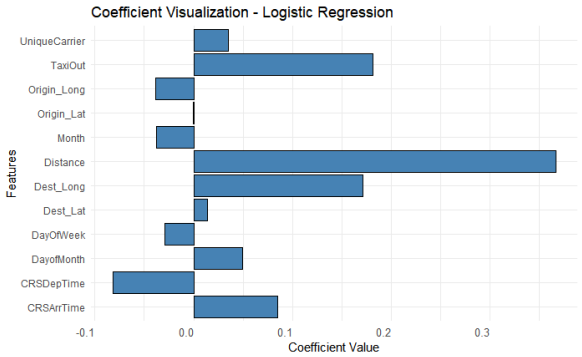




Python

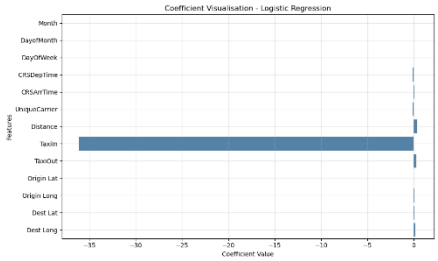


R



NOTE: It was observed that the 'TaxiIn' feature contributed to a drastic increase in the overall accuracy of the model from 0.66 to 0.96 and a slight increase in the f1-score. However, during coefficient visualization, it was uncovered that the coefficient value for 'TaxiIn' was an extreme outlier in comparison to other features. Therefore, the feature was not included in the model as it induced suspicions of bias due to the extreme coefficient values displayed for both years. While one may dispute the exclusion of the feature from the model, it is imperative to consider the broader implications of introducing a potentially biased feature. As depicted in the diagrams below (Python), the feature 'TaxiIn' is indeed an extreme outlier.

2006



Visualizing coefficients - TaxiIn included

2007

While increasing accuracy is desirable, it is also essential to ensure that the model stays unbiased and reliable. Therefore, the model was created excluding the outlier and the sub-par accuracy and f1-score of the model was attributed to the major sample discrepancy in the dataset for Diverted flights.