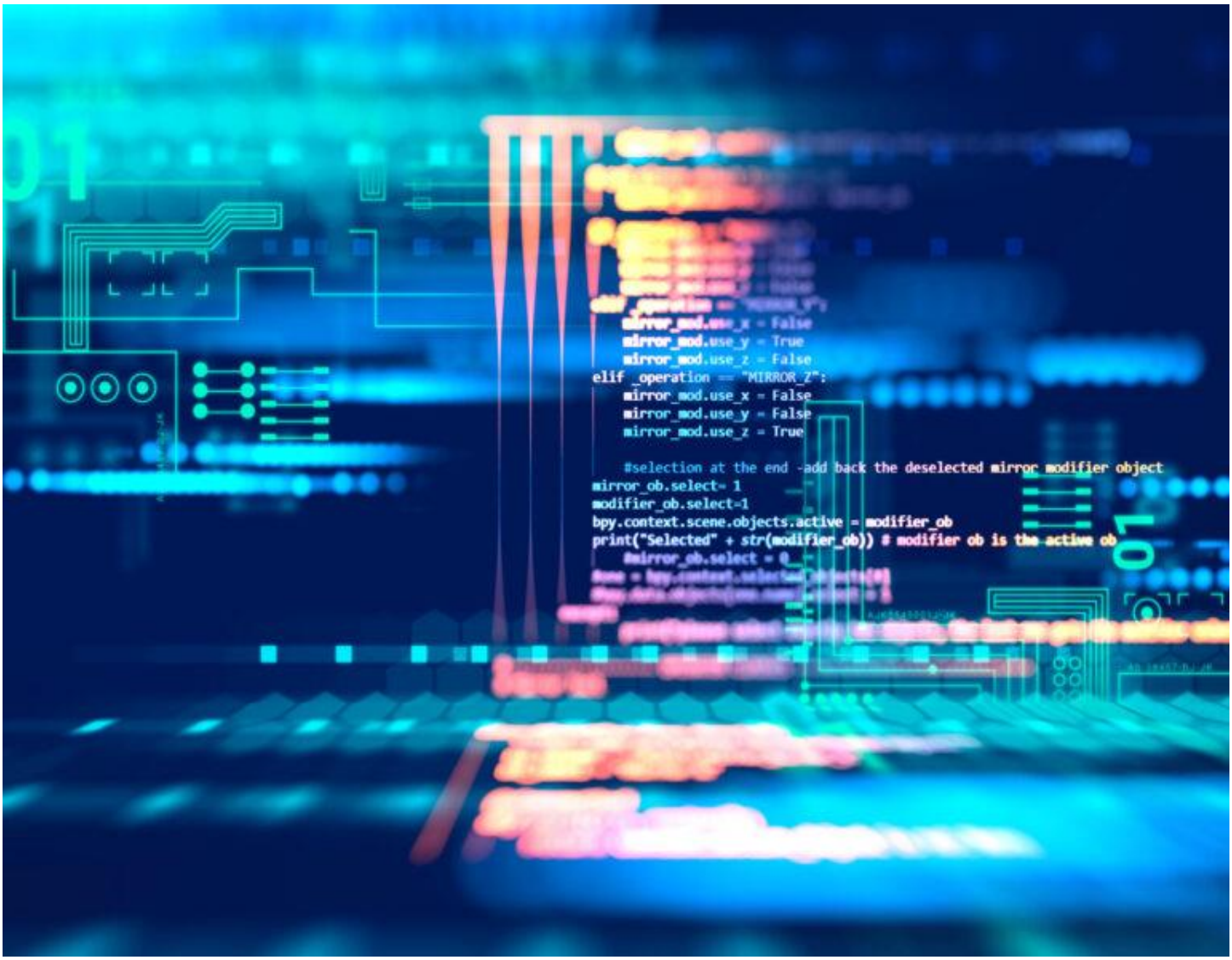
UNIVERSITY OF LONDON

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE



ST2195 COURSEWORK - REPORT

STUDENT NUMBER: **220642374**

PAGE COUNT: 7

Table of Contents

Part 1 (a)

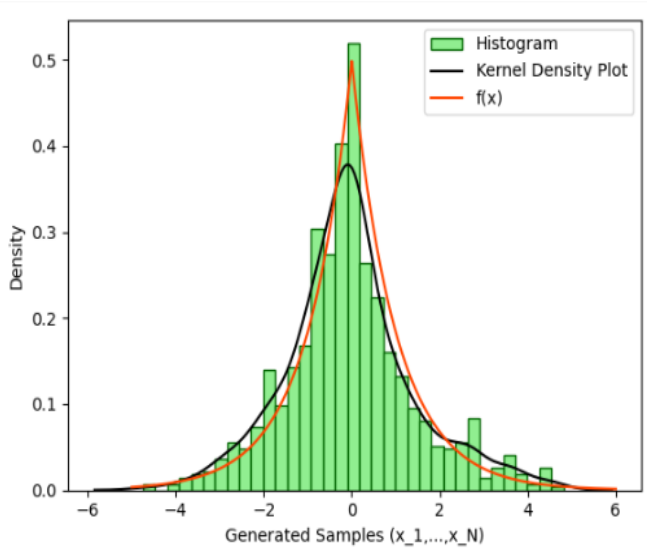


Figure 1 – Python

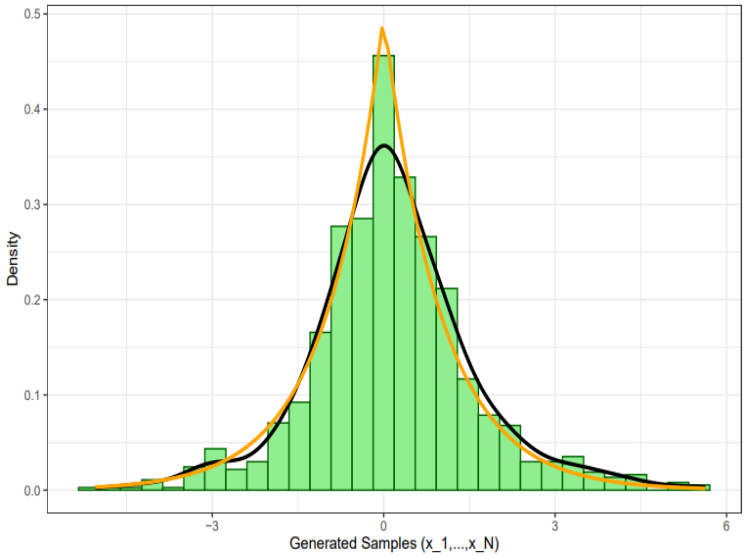


Figure 2 – R

Figure 1 - Generated samples obtained by applying the random walk Metropolis algorithm were used to construct a histogram and a kernel density plot. A graph of f(x) was overlaid on top.

The sample mean and standard deviation of the generated samples in python are -0.05099 and 1.46429 respectively to 5 decimal places and the sample mean and standard deviation of the generated samples in R are 0.1378865 and 1.418515 respectively.

Part 1 (b)

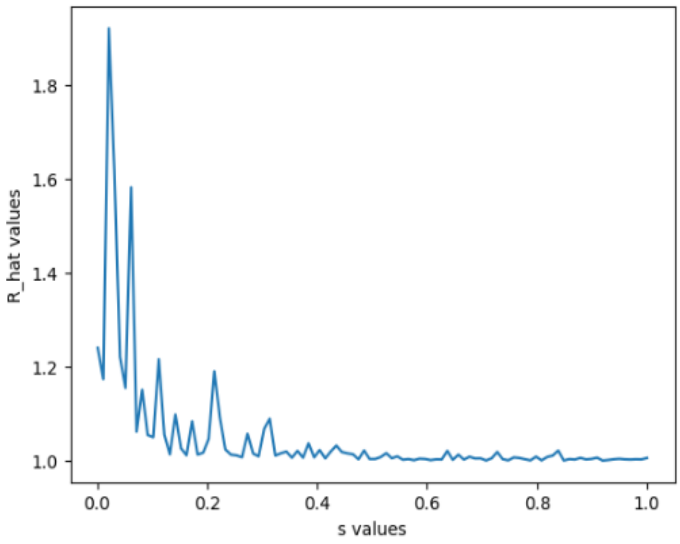


Figure 2 - Python

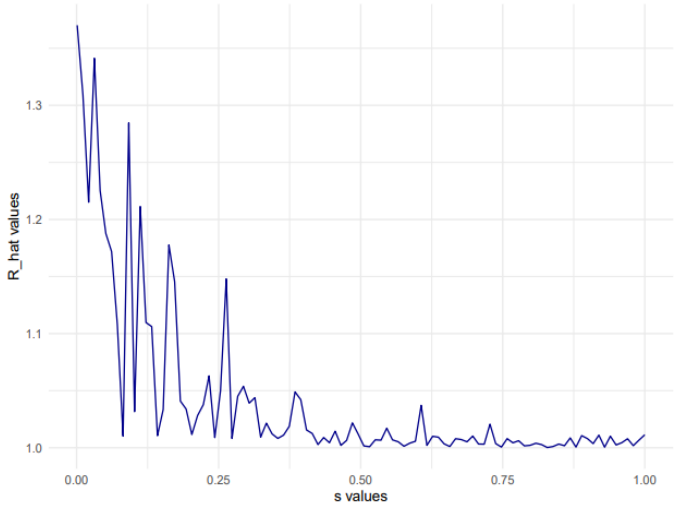


Figure 2 - R

Figure 2 – Plot of R hat values for a range of s values in between 0.001 and 1

```
R hat value when s is 0.001= 1.1253074725715657
```

```
[1] "R hat value when s is 0.001 = 1.20327072757766"
```

The R hat value when s is 0.001 is 1.1253074725715657 and 1.20327072757766 in python and R respectively.

Data Cleaning

The latest three years of data available (2006,2007 and 2008) were loaded. It was identified that 2008 dataset had only data till the month of April and only contained 2389211 observations, whereas 2007 and 2006 datasets had 7453211 and 7141918 observations respectively. Hence, the 2006 and 2007 datasets were merged to form one dataset. The data types were checked and 34 duplicated rows were identified and removed. The columns of the dataset were checked for null values and the 'CancellationCode' column had 14312420 null values and so the 'CancellatonCode' column was removed from the dataset. Only flights with 'DepTime' and 'ArrTime' within 24 hours of the day (that is below 2400 hhmm) were considered as inliers and so a constraint was implied. After the implication of the constraint automatically all the null values from the other columns had been removed and hence our dataset did not contain any other null values. A new column 'TotalDelay' was created by adding 'DepDelay' and 'ArrDelay'. The dataset was then saved as "**cleaned_dataset_2006_2007.csv**".

Part 2 (a)

Assumption : In this question time was assumed to be the CRSDepTime.

The "cleaned_dataset_2006_2007" was loaded as 'df' and used to answer this question.

The 'df' dataset was divided into separate subsets for each year because the question seeks to identify the optimal times and the optimal days of the week to reduce delays for each year individually.

The 'CRSDepTime' was divided into 6 equal time slots; 12am-4am, 4am-8am, 8am-12pm, 12pm-4pm, 4pm-8pm and 8pm-12am. Then the dataset was grouped by time slots and years to calculate the average total delay, ('AverageTotalDelay_TS') which was computed by taking the mean of the 'TotalDelay'. The 'AverageTotalDelay_TS' rounded to the nearest minute for simplicity and clarity as the differences were very small.

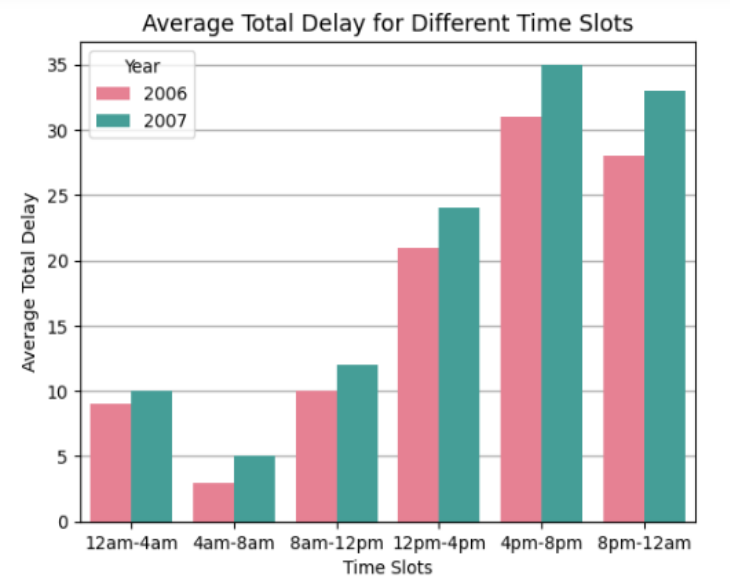


Figure 3 - Python

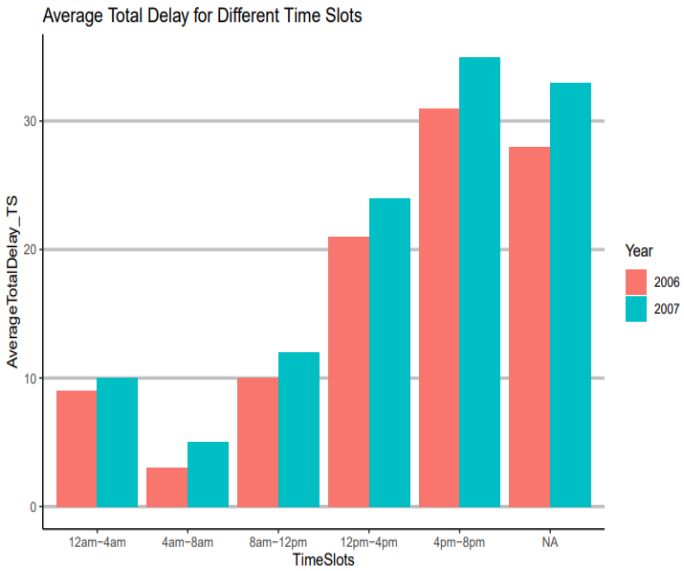


Figure 3 - R

Figure 3 – side by side bar plots for different time slots and years

From the visualization it can be identified that **planes which were scheduled to departure between 4am to 8 am have the least delay in both years**. The average total delay between 4am to 8 am was approximately 3 minutes in 2006 and 5 minutes in 2007.

The dataset was then grouped again by 'DayOfWeek' and 'Year' to calculate the average total delay, 'AverageTotalDelay_Dow' which was computed by taking the mean of 'TotalDelay'.

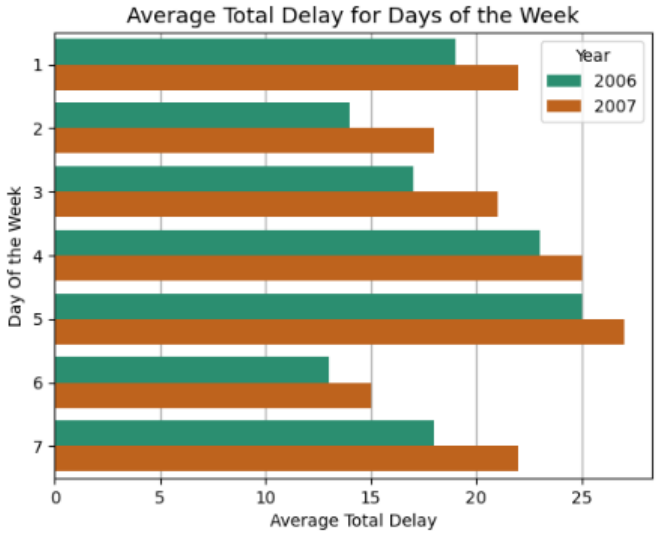


Figure 4 - Python

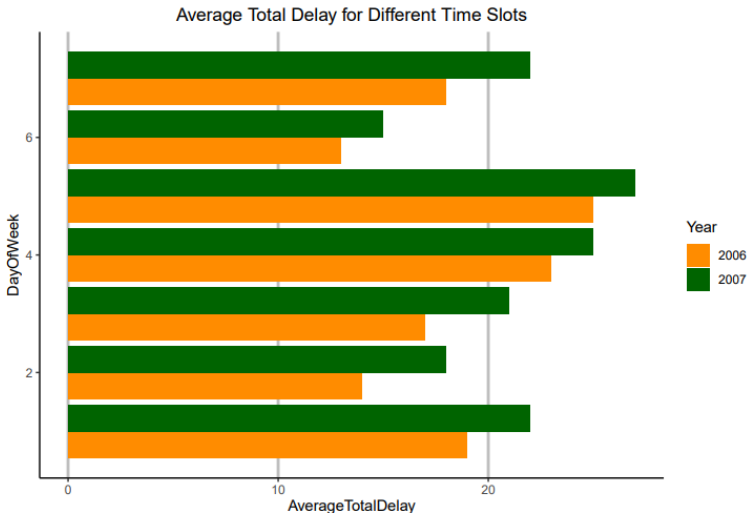


Figure 4 - R

Figure 4 – Side by side horizontal bar plots representing the average total delay for different days of the week by years

From figure 4 it can be diagnosed that the **6th day of the week (Saturday) has the least delay in both years,** which was 13 minutes in 2006 and 15 minutes in 2007 approximately.

Part 2 (b)

The "cleaned_dataset_2006_2007" was loaded as 'df' and used to answer this question.

The supplementary plane-data was loaded, analyzed and checked for missing values. Then the necessary columns, 'tailnum' and 'year' was filtered from the dataset. The 'year' column actually represent the year of manufacture of the planes therefore it was renamed as 'YearOfManufacture' and the 'tailnum' was renamed to 'TailNum' because it was the only common column in both the 'cleaned_dataset_2006_2007' and the 'plane_dataset' and hence is vital to have the same column name in both the dataset to perform inner merge.

The columns 'TailNum' and 'TotalDelay' was filtered from df and the merged with the filtered columns from the plane_dataset based on the 'TailNum' column. The merged dataset was stored as 'plane_df_merged' , and when checking for null values it was detected that the YearOfManufacture column had 692306 null values which was approximately only 0.05% of the data so the rows containing the null values were removed.

Two anomalies were spotted in some rows as it contained 0 and None under the 'YearOfManufacture' column which were not ideal hence those rows were removed.

The merged dataset was then grouped by year of manufacture and the average total delay for each year of manufacture was calculated.

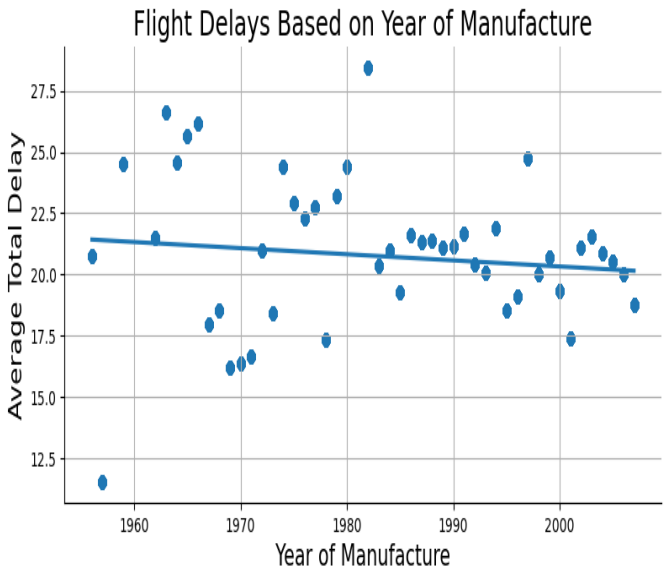


Figure 5 - Python

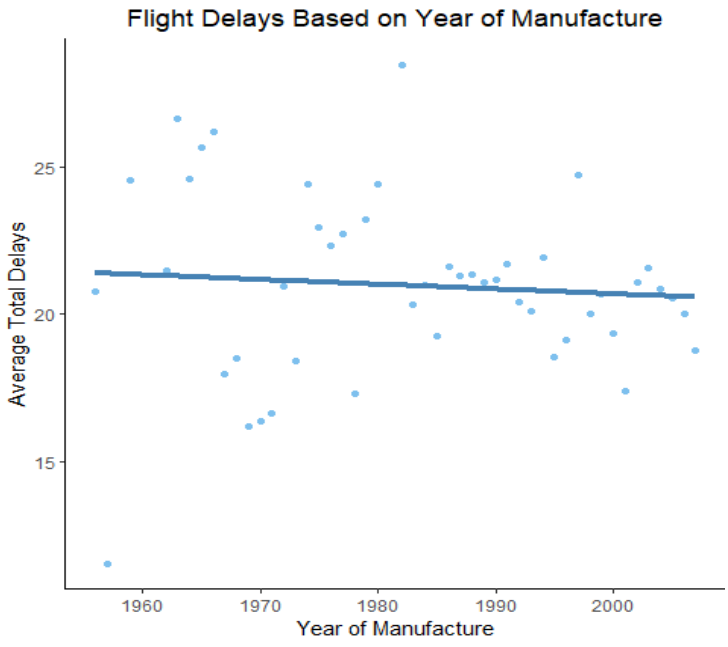


Figure 5 - R

Figure 5 – A scatter plot overlaid by a liner regression line

From figure 5 it can concluded that **older planes suffer more delays** than the new planes.

Part 2 (c)

Unfortunately the cleaned_datset_2006_2007 has a very high class imbalance as there are no data of diverted flights, this could have occurred due to the cleaning process hence the 2007 dataset was used to answer this question.

The 2007 dataset was loaded as ‘df2007’ and the diverted class was highly imbalanced as it contained 7436036 rows of not diverted flights and only 17179 of diverted flights.

The ‘df2007’ dataset was checked for null values and the unwanted columns were dropped.

The supplementary airport dataset was loaded and checked for the columns with null values then the unwanted columns; 'city', 'state', 'country’ and ‘airport' were removed. Assuming that the iata column in the airport_dataset represented the Origin it was renamed as ‘Origin’.

Afterwards the two datasets were inner merged based on the ‘Origin’ column

The correlation matrix was created to show the correlation between the columns of the merged dataset

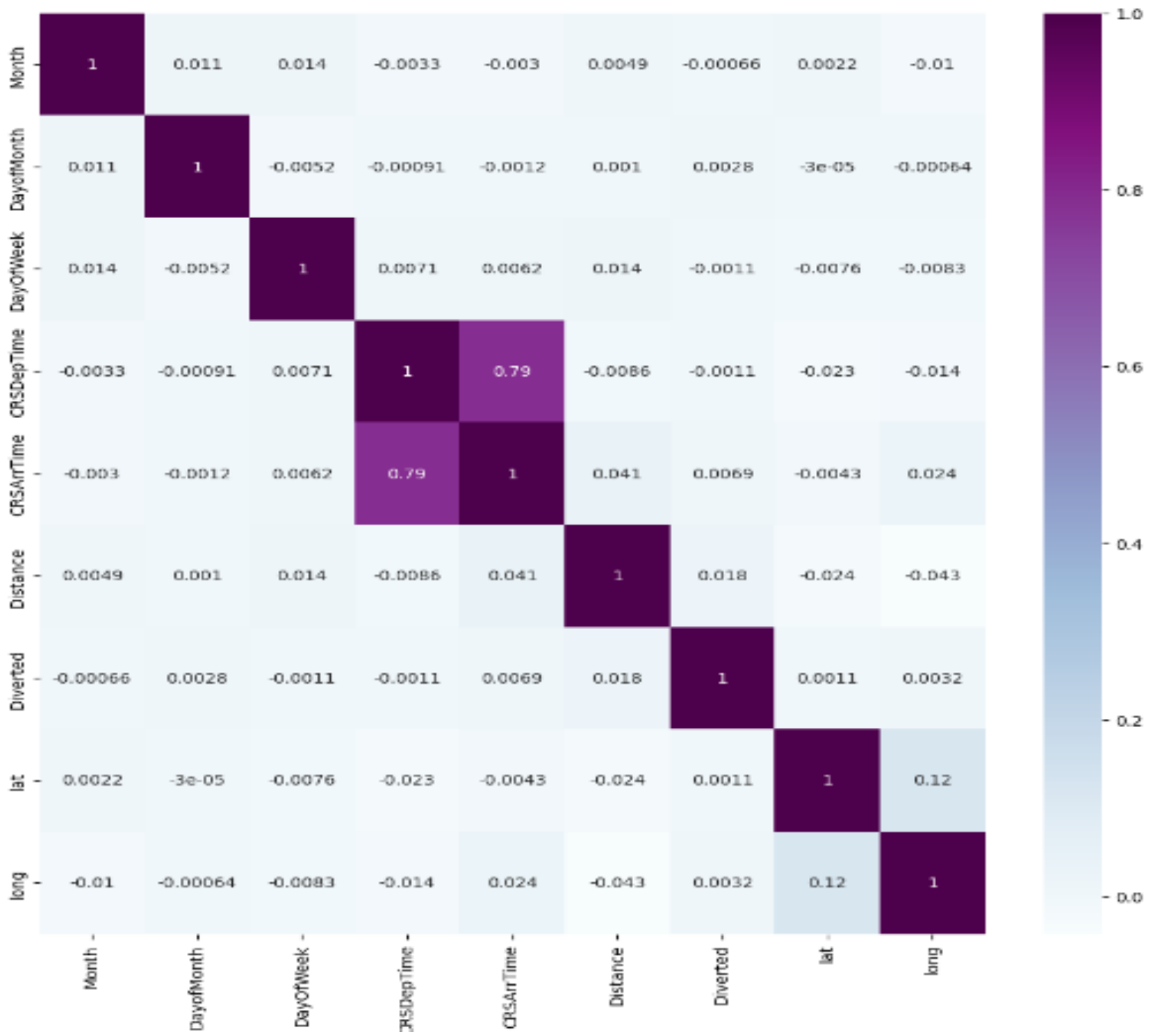


Figure 6 - Python

'Month' , 'DayofMonth' , 'DayOfWeek' , 'CRSDepTime' , 'CRSArrTime' , 'Distance' , 'lat' , 'long' were the features selected and the target variable was 'Diverted'. The train and test was carried out. The logistic regression model was applied to predict the coefficients of the features.

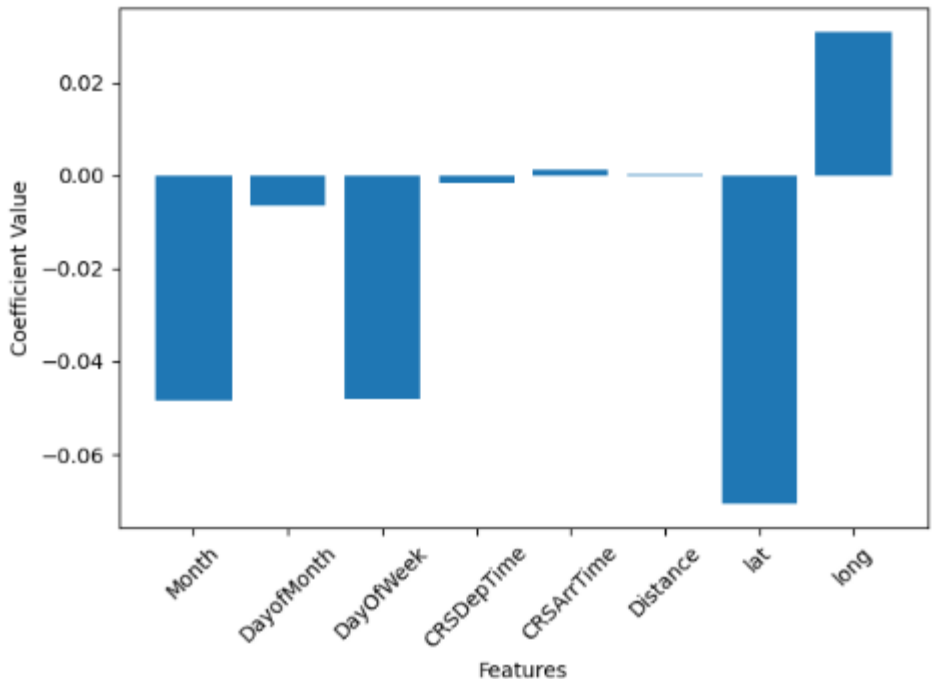


Figure 7 – python

From figure 7 it can be concluded that the longitude has a positive relationship with flights being diverted or not and the latitude has a negative relationship. We can see that the CRSDepTime ,CRSArrTime and the distance doesn't really affect the flights to be diverted or not. The month and the DayofWeek also have a negative coefficient impling an inverse relationship with flights diversion.