UNIVERSITY OF LONDON

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

ST2195 – Programming for Data Science

(Coursework Report – Python and R)

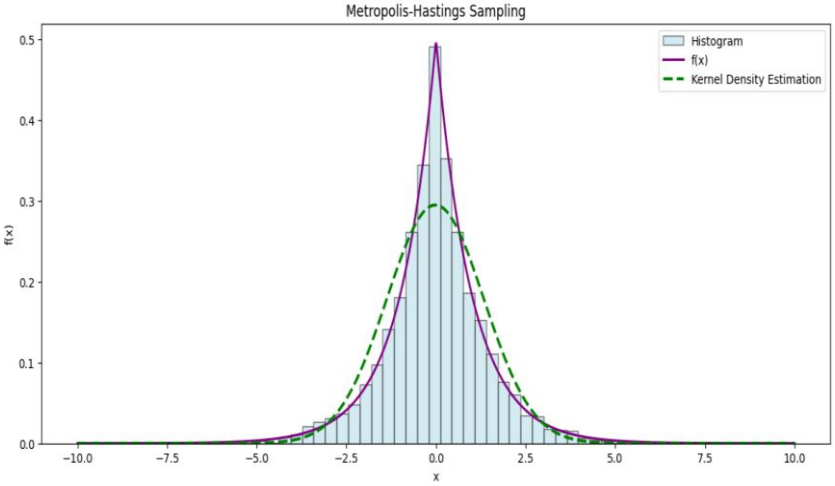
UOL Student ID - 220627669

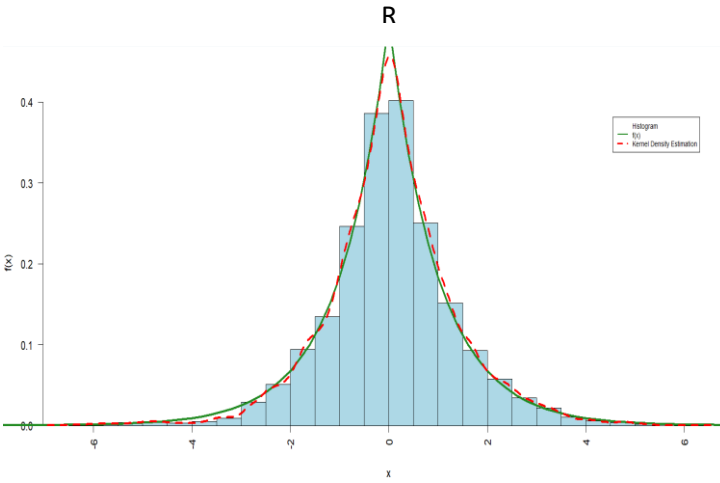
Page Count – 7 (Excluding cover page and table of contents)

Table of Contents

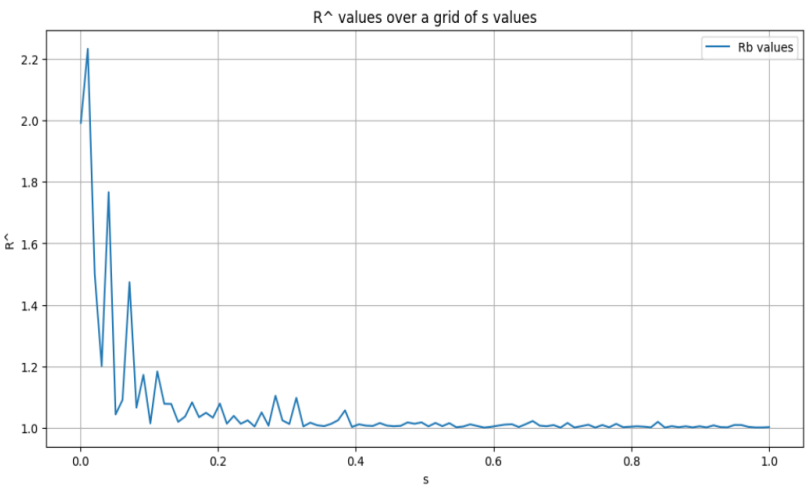
PART 1

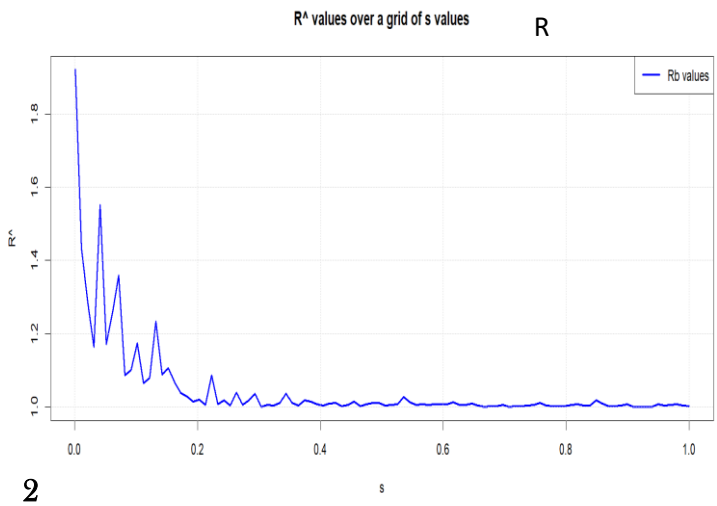
(a) The Metropolis-Hastings algorithm is used to simulate random numbers from complex probability distributions. The target probability density function was defined as f(x)=0.5exp(|x|), representing a distribution that decays exponentially as |x| increases. To apply the algorithm, parameters were set, with the number of samples to be generated(N) = 10000, the standard deviation (s)= 1 and the initial value (x0) =0. Hence for each iteration from 1 to N (10,000 times), a new sample x* was proposed from a Normal Distribution, centred at the current sample x(i-1) and standard deviation s. Then the acceptance ratio r= f(x*)/ f(x(i-1)) was calculated, which determines the likelihood of accepting the proposed sample based on the target distribution f(x). Next, a random number 'u' is generated from a Uniform Distribution, between 0 and 1. If u<r, the proposed sample x* is more probable under f(x) compared to the current sample x(i-1). Thus, we accept the proposed sample and set xi = x*. Otherwise, xi = x(i-1). A histogram with 40 bins is drawn to illustrate the density of samples across a range of x. It is overlaid with the sample pdf f(x) for visual comparison. The Kernel Density plot, computed using the normal distribution, is a smooth curve and aligns closely with f(x), which indicates the effectiveness of the sampling algorithm. The sample mean = -0.0195 is close to zero, which aligns with the shape of f(x). The sample standard deviation = 1.35 is relatively large and indicates variability in the sample distribution. Hence, in conclusion, the Metropolis-Hastings algorithm has effectively explored the target distribution and generated samples that capture its central tendency while also reflecting its variability.





(b) The Gelman-Rubin statistic (R^) is employed to evaluate the convergence of the Metropolis-Hastings algorithm, by running multiple chains with different initial values (to ensure the reliability of the generated samples). First, a function is defined to compute the R^ statistic. It calculates the sample mean (Mj) and within-sample variance (Vj) for each chain. It then computes the overall within-sample variance (W), overall sample mean (M) and between-sample variance (B). Finally, it calculates the R^ statistic using the formula (B+W/W) ^0.5. Next, a function is defined to execute multiple chains of the Metropolis-Hastings algorithm and stores the generated chains in an array. It generates J=4 chains, each consisting of N= 2000 iterations. Chains are initialized with an initial value x0=0 and standard deviation s=0.001. A range of s values from 0.001 to 1 is then generated, and the R^ value is calculated for each s value. In order to visualise how the convergence varies with different step sizes, the R^ values are plotted against a grid of s values. The calculated R^ values using the initial parameters (N=2000, s=0.001, J=4) is 1.1875. This suggests potential lack of convergence, as it exceeds the desirable threshold of 1.05 (ideally, lower R^ values closer to 1 indicate better convergence). Through the graph, it can be observed that for certain (larger) s values, R^ is closer to 1 and depicts better convergence. Therefore, based on the analysis, adjusting/increasing the step-size s could potentially improve convergence and increase the reliability of the generated samples from the Metropolis-Hastings algorithm, as indicated by variations in the R^ values.





PART 2

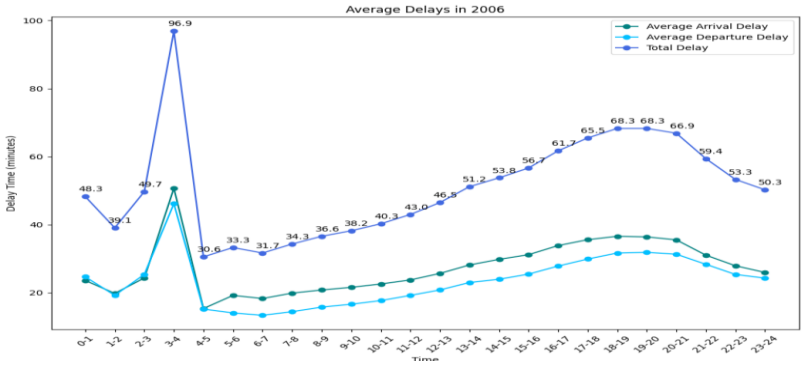
Introduction - This report is based on a subset of the 2009 ASA Statistical Computing and Graphics Data Expo, which includes flight arrival and departure details for major carriers within the USA from October 1987 to April 2008. The datasets used in this project are from the years 2006 and 2007, and additional CSV files on airports, plane data and carriers were also utilised for parts (b) and (c).

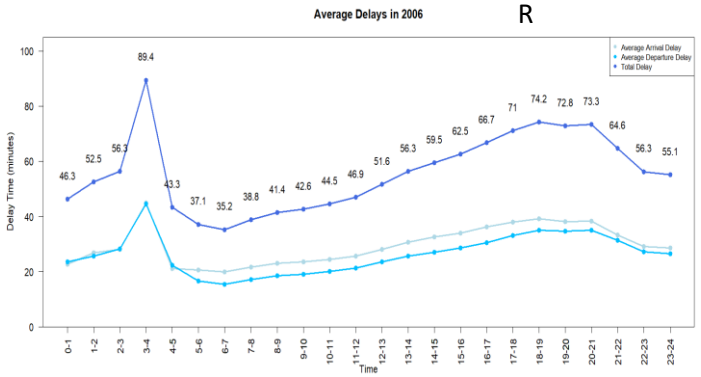
Data Cleaning Process - As the columns of both the 2006 and 2007 year datasets provided to us were the same, they were combined to form a single dataset. The duplicated rows were removed initially, and Null Values were checked for. The 'CancellationCode' column was removed entirely due to meaningless values (either NA or 0), and missing values were added to the 'CRSElapsedTime' and 'TailNum' columns by substituting them with the median and mode of each column respectively. A new column called 'Total_Delay' was created in the dataset by adding 'ArrDelay' and 'DepDelay' columns. The merged dataset was then exported to a CSV file named **cleaned_dataset.csv** to answer future questions. The cleaned dataset was then split into two separate data frames containing 2006 and 2007 data only, labelled 'delays2_2006' and 'delays2_2007' respectively. In order to analyse arrival and departure delays separately, these 2 datasets were further split into data frames called 'arrival_delays2006', 'departure_delays2006', 'arrival_delays2007' and 'departure_delays2006' which contain columns with arrival and departure data separately. Null values were dropped from 'ArrTime', 'ArrDelay', 'DepTime' and 'DepDelay' columns, and then negative or 0 values in the 'ArrDelay' and 'DepDelay' columns were dropped as well. This is because delays<0 represent early arrivals and departures, delays=0 indicate that the plane has landed/departed on time and delays>0 means that there has been a delay (which is what we want to analyse). The Delay time is given in minutes.

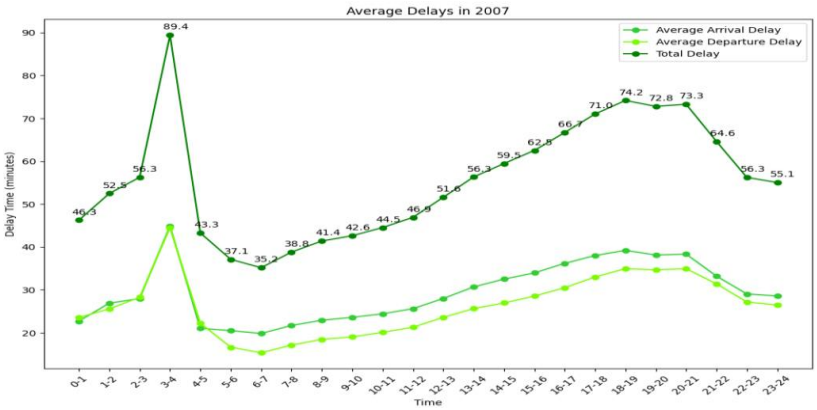
(a) What are the best times and days of the week to minimise delays each year?

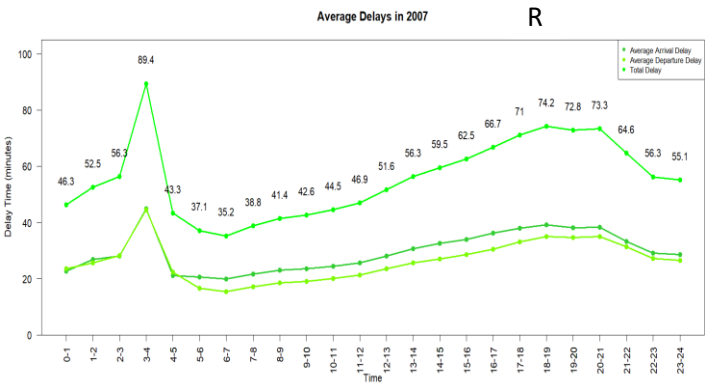
Best Time of Day to minimise Delays

The hours from 'CRSDepTime' column are extracted and are divided by 100 to get the hour part only. It removes any minutes/ seconds present in the time, bins the scheduled departure time by hour and adds it to a new column named 'Hours_Binned'. Range and labels are defined for 24 hours of the day, and the average arrival and departure delays are calculated separately for 2006 and 2007, grouped by CRSDepTime. Total average delays are calculated by adding average arrival and departure delays for 2006 & 2007 separately. A line graph is created to display data:





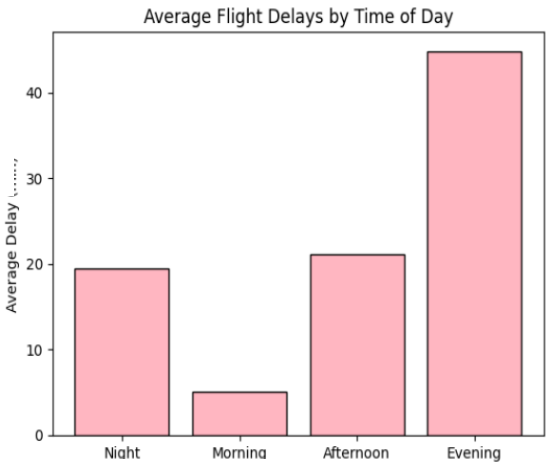




In 2006, it can be observed that 04.00-05.00 have the least total delays and the highest delays are from 03.00-04.00. Least arrival delays are from 04.00-05.00 and the least departure delays are from 06.00-07.00. In 2007, the least total delays are from 06.00-07.00 and the longest delays are again at 03.00 -04.00 (both years have a steep drop in delays from 04.00-05.00). In both years, the delay time increases from 07.00 to 20.00 for arrival and departure delays (most delays are from 18.00 to 21.00).

To calculate the overall average delays for both 2006 and 2007, the departure time was binned into 4 time slots with a 6hr difference. ('Morning' : 06.00-12.00, 'Afternoon' : 12.00-18.00, 'Evening' :18.00-24.00, 'Night': 24.00-06.00)

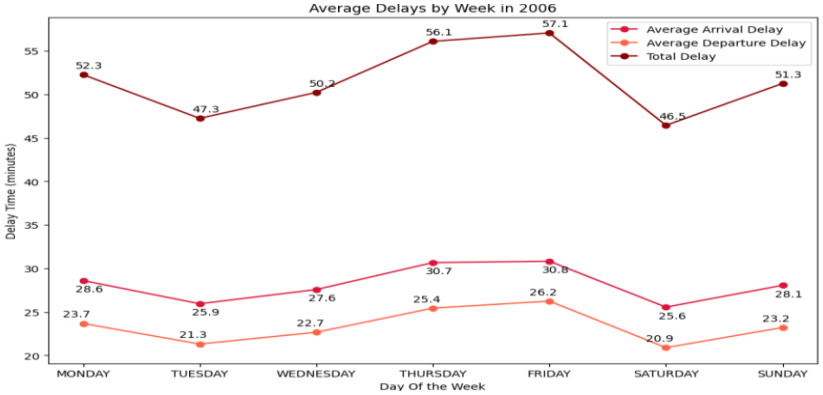
Average arrival, departure and total delay (for 2006 & 2007 combined) was calculated for each time slot and displayed in a table.
It is evident that morning (00.00-06.00) is the best time of day to minimise delays, and evening (18.00-24.00) should be avoided as it contains the longest arrival, departure and total delays.



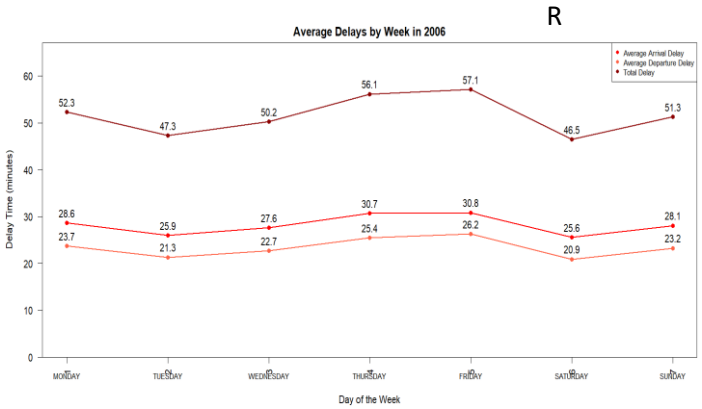
	Time Of Day	Avg Arrival Delay	Avg Departure Delay	Avg Total Delay
0	Night	8.843243	10.604970	19.448214
1	Morning	1.877135	3.231844	5.108978
2	Afternoon	10.038824	11.112748	21.151572
3	Evening	21.705334	23.108350	44.813684

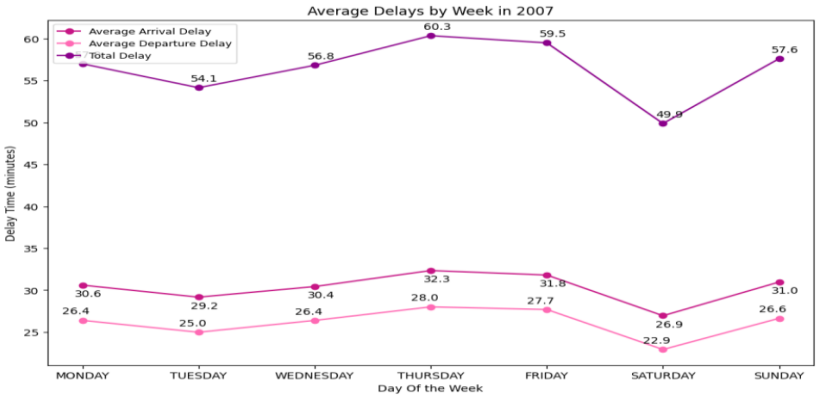
Best Day of the Week to minimise Delays

The 'DayOfWeek' column was replaced with strings 'Monday, Tuesday…Sunday' for easier readability. The average arrival and departure delays by week were calculated separately for 2006 & 2007, grouped by the 'DayOfWeek' column. The total delay by week for each year was calculated by adding the average arrival and departure delays by week. A line graph is plotted for each year to show the average delays for each day of the week.

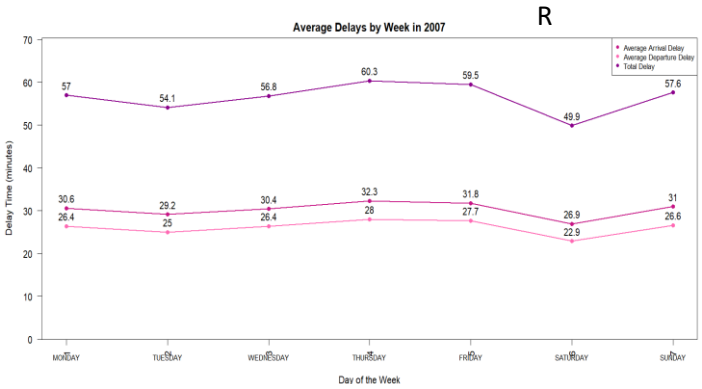


R





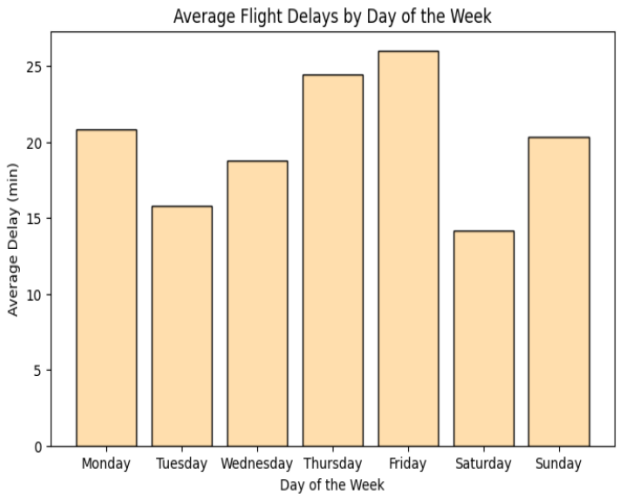
R



It can be seen that the least arrival and departure delays both occur on Saturdays in 2006, whereas the highest delay times are on Fridays . In 2007, the least total delays also occur on Saturdays, but the longest delays are on Thursdays.

In order to calculate the overall average delays by week (for both 2006 & 2007), first the missing values in 'ArrDelay' and 'DepDelay' were filled with zeros. Then the average arrival, departure delays and total delays were calculated, grouped by the 'DayOfWeek' column. A table was created by merging data frames containing the delay values and days of the week and a bar chart was used to visualise the average flight delays by day of the week, for both 2006 and 2007.

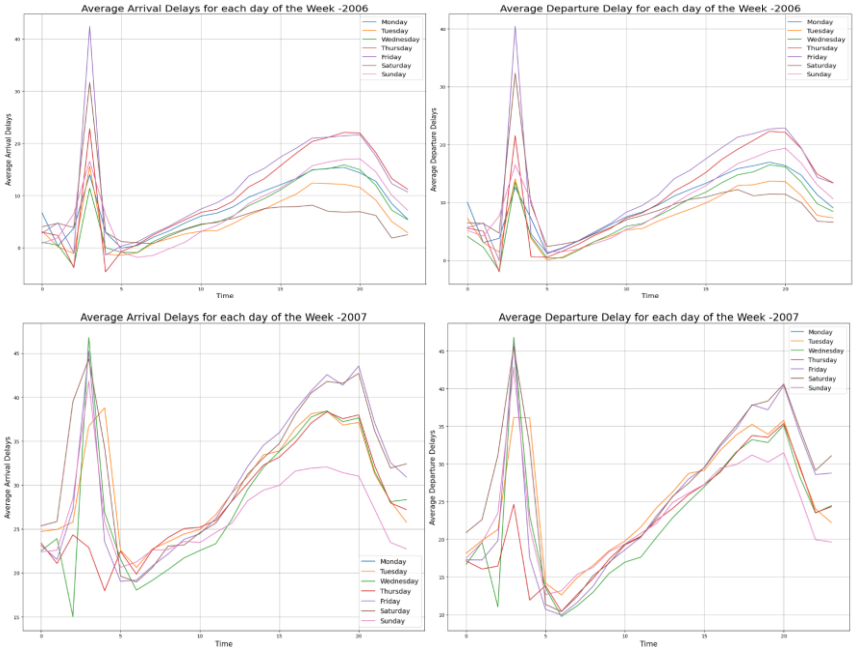
	DayOfWeek	Avg Arrival Delay	Avg Departure Delay	Avg Total Delay
0	Monday	9.701637	11.106441	20.808078
1	Tuesday	7.248411	8.532385	15.780797
2	Wednesday	8.987954	9.771908	18.759863
3	Thursday	12.141524	12.314941	24.456465
4	Friday	12.701676	13.299041	26.000717
5	Saturday	5.513355	8.620817	14.134172
6	Sunday	9.247222	11.074626	20.321849

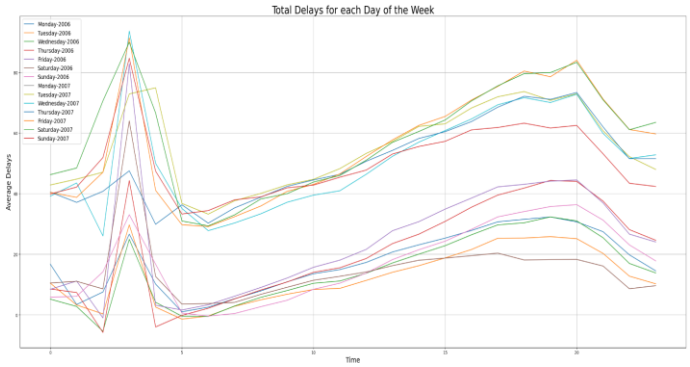


Hence it is clearly seen that Saturdays are the best day of the week to minimise delays. Friday and Thursday have the highest delays and therefore must be avoided.

Best Time of the Week to minimise Delays

The 'delays2_2006' data frame was copied so that its changes don't affect the original dataset, and the column 'Hours_Binned' was added to it from the dataset 'arrival_delays2006'. In order to apply the 'pd.to_datetime' function, the column 'DayOfMonth' was renamed to 'Day'. Then the 'DayOfWeek' column was converted to the datetime format and the day of week (eg: 0 for Monday) was extracted.





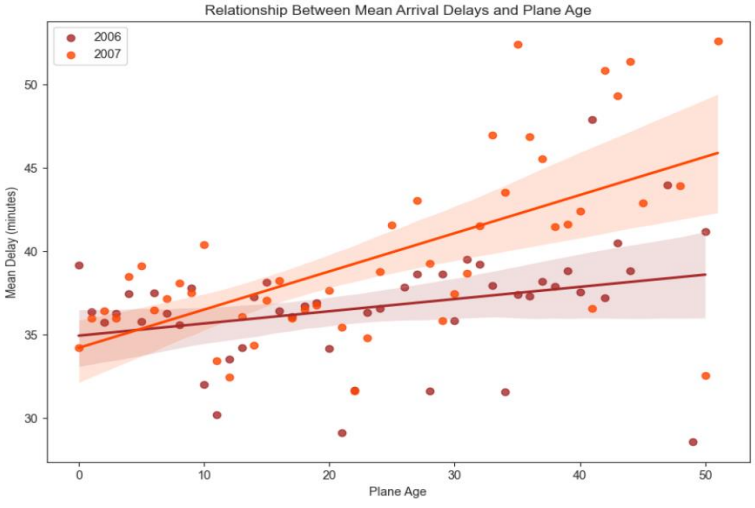
To extract the hour of day, the 'CRSPDepTime' column was binned by dividing it by 100 (to remove the minutes and seconds) and assigning the integer part to the 'Hours_Binned' column. The average arrival and departure delay, and the total delay for each hour of the day was then calculated, grouped by the 'DayOfWeek' and 'Hours_Binned' columns.

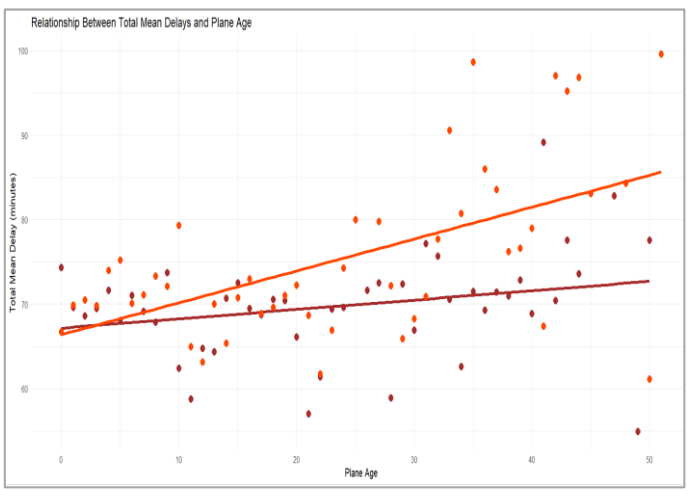
This process was then repeated for the 2007 data, using the 'delays2_2007' and 'arrival_delays2007' data frames.
The first four graphs represent the average arrival/departure delays at each time on each day of the week, for years 2006 and 2007. The last graph shows the total delays at each time of the week for both years together.
It can be seen that there is a high peak at 03.00 for each day of the week (in all the graphs) which means maximum delays occur daily at that time. There are also high delays every day from 15.00 to 20.00 in both 2006 and 2007. The minimum delays occur daily between 04.00 – 06.00 (after a large drop in delays from 03.00). Hence it can be stated that 04.00-06.00 are the best times of the week to avoid delays.

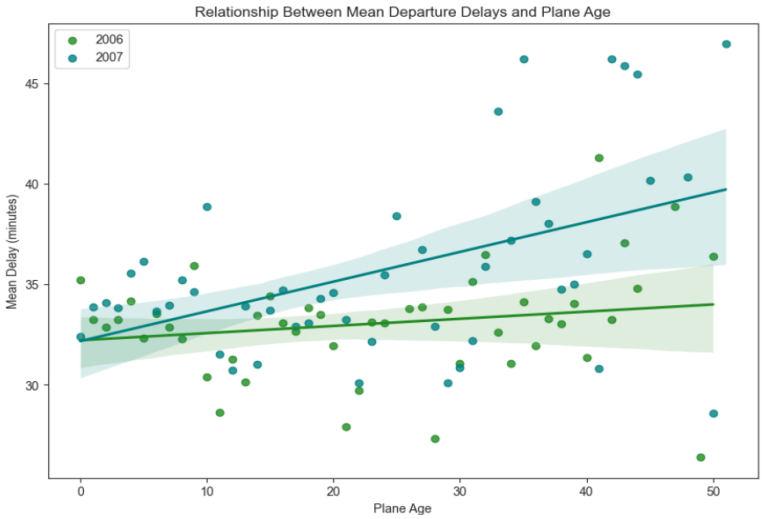
(b) Evaluate whether older planes suffer more delays on a year-to-year basis.

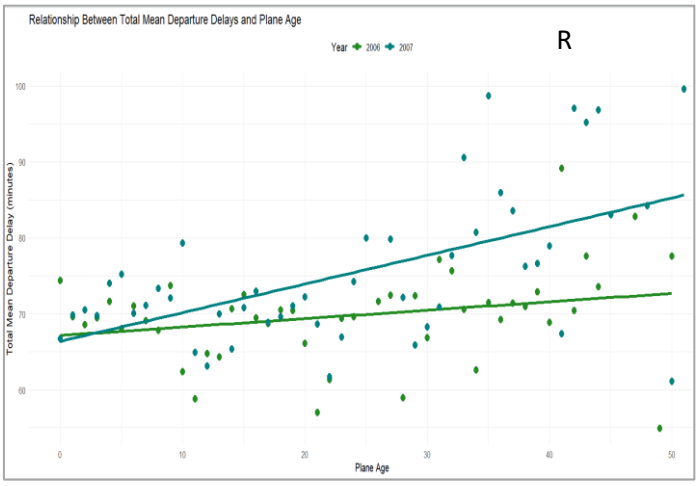
This analysis is done for both Arrival and Departure delays, as well as Delay Types to check the relationship between Plane Age and Delays. The plane data set was imported and cleaned, by first converting the 'None' data into NaN values. Then all NaN (missing) values were dropped, as well as the duplicated rows. The column 'tailnum' was renamed as 'TailNum' and the plane-data was left-merged with the cleaned_dataset on 'TailNum' to create a new data frame 'merged_with_planes'. The column 'year' was renamed to 'YearOfManufacture' and then the required columns for (b) were extracted into a data frame 'planes_b'. Null values were dropped, the data type of 'YearOfManufacture' was converted to Integer and all rows with the value 0 were removed. 'PlaneAge' column was calculated by taking the difference between 'Year' and 'YearOfManufacture' columns. 'PlaneAge' values <0 were converted into null values and then dropped, and all the early arrivals and departures ('ArrDelay' <0 and 'DepDelay'<0) were removed as well. Finally, the dataset was spilt into 2006 and 2007 data, and the columns required for arrival and departure delays were extracted separately into 4 different data frames. 'PlaneAge' values were converted into Integers and then the mean arrival and departure delays were calculated for 2006 and 2007, grouped by 'PlaneAge'.

R

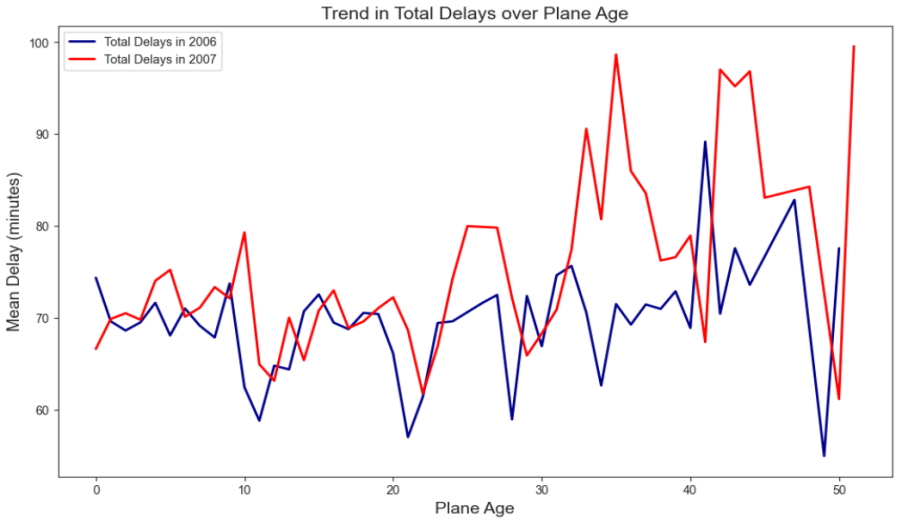




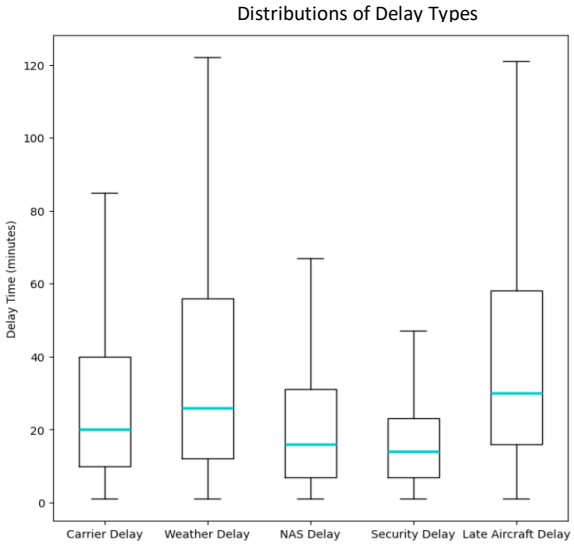




The calculated correlation of Mean Arrival Delay and Plane Age in 2006 is 0.3045 and in 2007 is 0.6274. Similarly, the calculated correlation of Mean Departure Delay and Plane Age in 2006 is 0.1902 and in 2007 is 0.4698. Hence it can be observed that there is a stronger positive correlation/relationship between Arrival Delays and Plane Age, compared to Departure Delays and Plane Age. So older planes suffer more Arrival Delays than Departure Delays. It can also be noted that all delays in 2007 have a stronger correlation with Plane Age than the delays in 2006, which could be an indication that planes experience more delays as they get older.

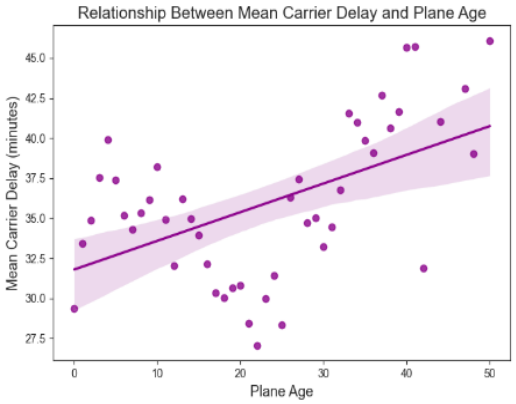


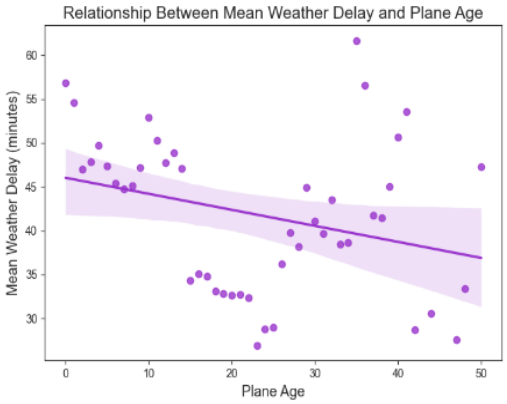
It can be seen that as the Plane Age increases, there is an overall rise in Total Delays for both 2006 and 2007. However, the sudden large drops and peaks in the delays indicate that there is no proper linear relationship between delays and Plane Age. Also, the delays in 2007 have clearly increased with Plane Age at a significantly higher rate than in 2006.

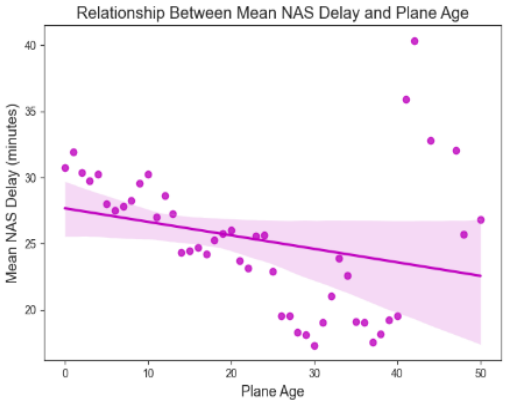


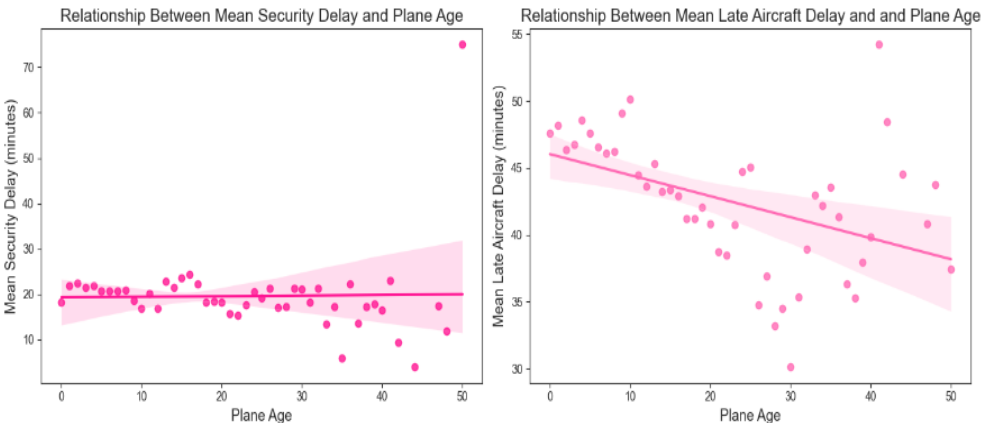
To check the relationship between Types of Delays and Plane Age, a new data frame 'delay_types' was created using the required columns of planes_b, including 'CarrierDelay', 'WeatherDelay', 'NASDelay',' SecurityDelay' and 'LateAircraftDelay'. All null values were removed and separate arrays were created for each Delay Type and the Plane Age. All the Delay values <1 were turned into null values and removed from the arrays, and boxplots were drawn to check their distributions.

Next, the mean delay of each Delay Type was calculated, grouped by the 'PlaneAge' column and this data was merged into a new data frame 'merged_delaytypes' in order to display the values in a table. Scatter plots were then drawn to display the relationships between the Delay Types and Plane Age.

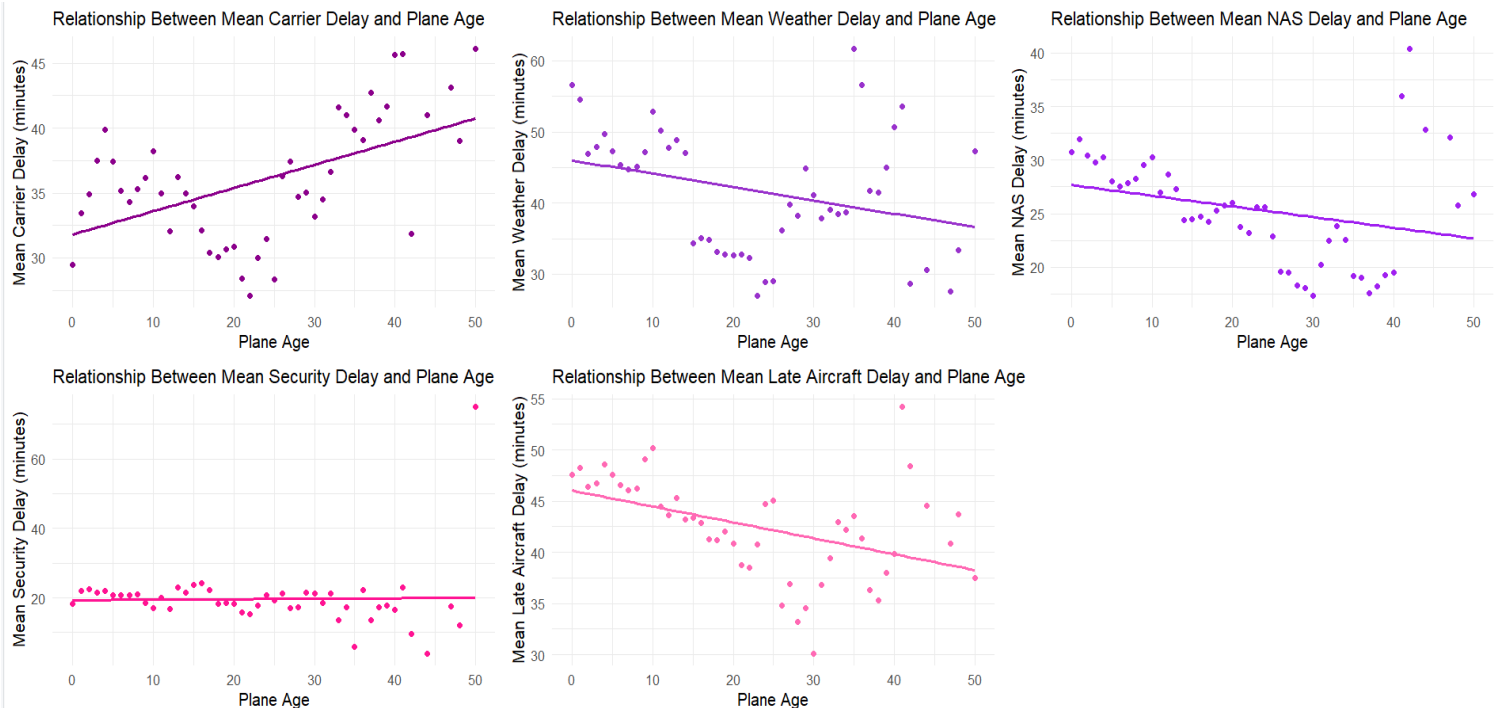








R



Carrier Delay &Plane Age correlation is 0.5124, Weather Delay & Plane Age correlation is 0.2915, NAS Delay & Plane Age correlation is 0.2953, Security Delay & Plane Age correlation is -0.0090 and Late Aircraft Delay & Plane Age correlation is -0.4505. It can be seen that Carrier Delay has a relatively strong positive relationship with Plane Age, while all the others have negative correlation coefficients. Late Aircraft Delay has the strongest negative relationship and Security Delay has a nearly zero correlation coefficient (no relationship with Plane Age).

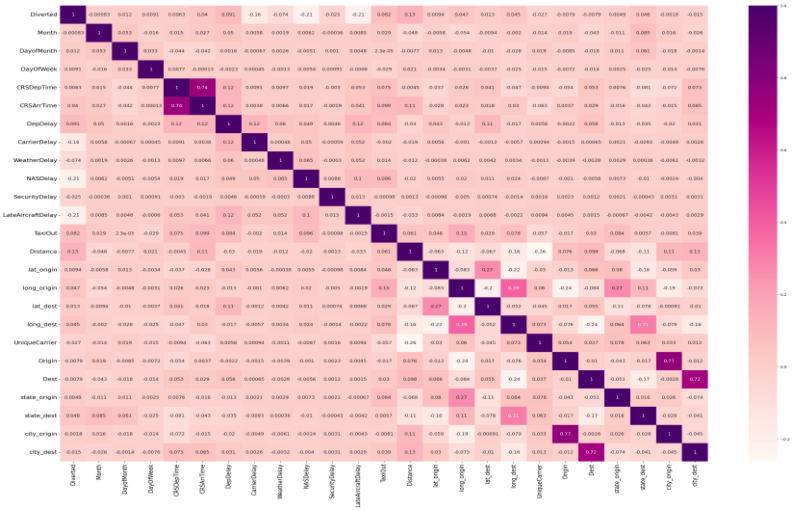
(c) Logistic Regression model for the probability of Diverted US flights.

For this part, the 'cleaned_dataset', 'carriers' and 'airports' datasets were used. The required columns were extracted from the cleaned_dataset and filtered to contain only 2006 and 2007 data separately. Next, the 3 datasets were merged together, by using the 'iata' column from airports dataframe, 'Origin', 'Dest' and 'UniqueCarrier' columns from the 2006/2007 dataframes and 'Code' from carriers dataframe. The column 'Description' was dropped and the null values were removed as well. The count of 0's and 1's in the 'Diverted' column were checked, where a huge imbalance was detected (7003802 zeros and 16186 ones).

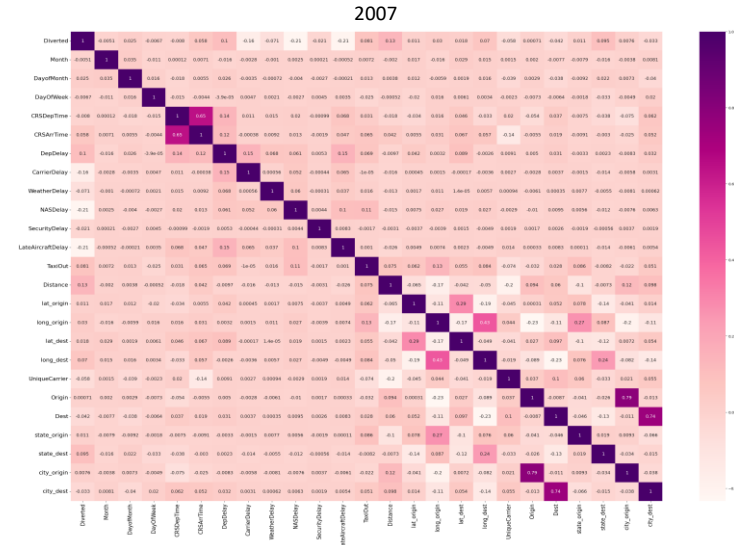
Next, the dataset was split into features(x) and the target variable(y) by dropping the 'Diverted' column . The train_test_split function was used to split the data into training and testing sets, allocating 20% of the data for testing and 80% for training. A bar plot was drawn to visualise the imbalance of 0's and 1's in the training set. The feature variables and target variables of the training set were then concatenated to produce the 'train_set' data frame. In order to balance the classes, the 'train_set' dataframe was divided into a majority class containing 0s and a minority class containing 1s. Using the resample function, the majority class was downsampled and the minority class was upsampled to balance each other, and then combined to create a balanced dataset. From this, the categorical columns to be used as attributes were selected and encoded using the LabelEncoder method, so that they can be used in future correlation and regression calculations. The numerical attribute columns were also selected separately and a new encoded dataframe 'df_2006' was made by joining the encoded categorical and numerical columns.

This was used to visualise the strength of the relationships between the 'Diverted' column and other attributes by drawing a correlation heatmap.

2006

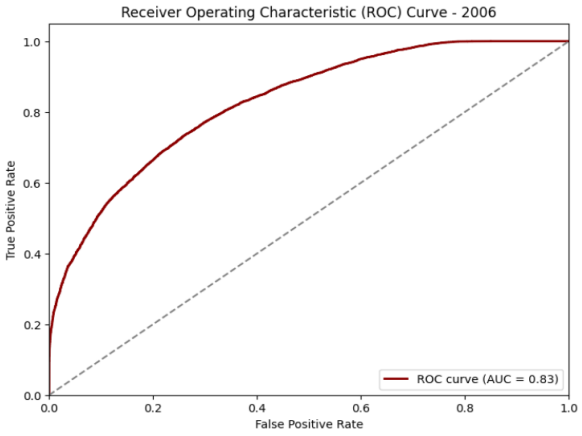


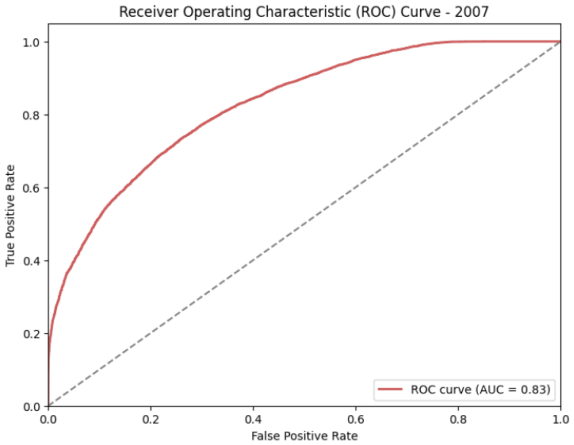
2007



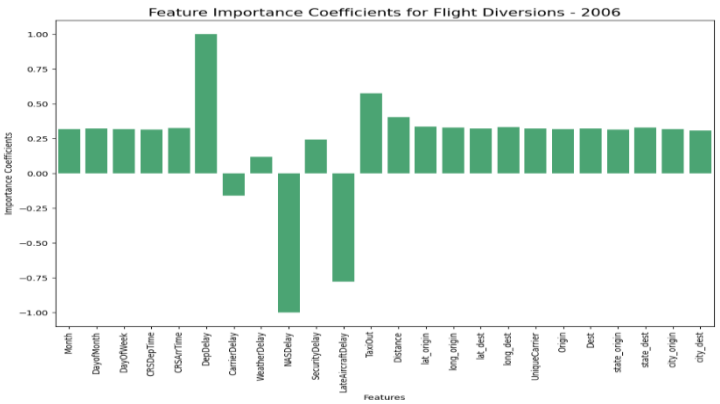
Distance and DepDelay have the strongest positive correlation with diversions, whereas LateAircraftDelay and NASDelay have the strongest negative correlation coefficients. Origin has the weakest overall correlation with the 'Diverted' column.

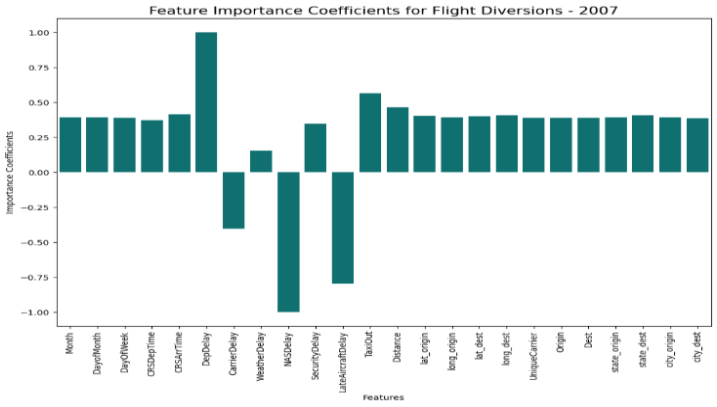
The 'Diverted' value counts are now much more balanced due to resampling and the column was then dropped and used to create new feature variables(x) and target variables(y). They were split once again to form new encoded training and testing data with test size of 0.2, and then standardized using StandardScaler to ensure that all features are on the same scale. A Logistic Regression model was fitted onto the training data, and labels were predicted for the testing data. The accuracy of the model on the test data is 0.883 in 2006 and 0.861 in 2007. Probabilities on the standardized test data were also predicted and then used to calculate the false positive rate (FPR) and true positive rate (TPR), as well as the F1 score. (F1 score for 2006 is 0.936 and 0.932 in 2007). An ROC curve was also plotted and had an AUC value of 0.801 in 2006 and 0.827 in 2007. These metrics show that the performance of the Logistic Regression model in terms of accuracy, precision and recall is good enough to predict the diversions in the US flights.





The importance coefficients of the features of flight diversions were then visualised in a bar graph, as shown below.





Hence, the significance of each feature in the Logistic Regression model can be observed, where Departure Delays, TaxiOut and Distance have the strongest positive relationships with Diversions, and NASDelay, LateAircraftDelay and CarrierDelay have the highest negative relationship. Suprisingly, WeatherDelays are the least significant attribute in this model and the remaining features have approximately the same positive effect on flight diversions in the US.