Awarding Body

UNIVERSITY OF LONDON

Academic Direction

LSE

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

ST2195 PROGRAMMING FOR DATA SCIENCE

COURSEWORK REPORT

UOL STUDENT NUMBER : 210468113

TABLE OF CONTENTS

INTRODUCTION

This report is based on planes that have flown over various areas over time, as well as the delays they have experienced when departing from and arriving at airports in the United States of America (USA). Because a lot of data is needed, particularly when developing machine learning models, the "2006" and "2007" year datasets obtained from the Harvard Website were found to be the cleanest datasets and also included the largest number of observations out of all datasets provided, and thus those were chosen to be explored in both R and Python programming languages. Moreover, while answering some questions, extra information was needed. Hence, csv files on airports and planes were used.

The report is divided into six sections:-

1. The data cleaning process that was implemented to make the data more useful and easier to analyse.
2. Finding the best time of day, day of week and time of year to fly to minimise delays.
3. A review into whether older planes are more likely to be delayed.
4. Analysing the fluctuations in the number of passengers flying between different areas over time.
5. Checking to see whether delays in the previous airport cause delays in the current airport.
6. Creating a model to predict arrival delay.

Each section includes graphs and tables created in R and Python that were useful in reaching the final result.

DATA CLEANING PROCESS

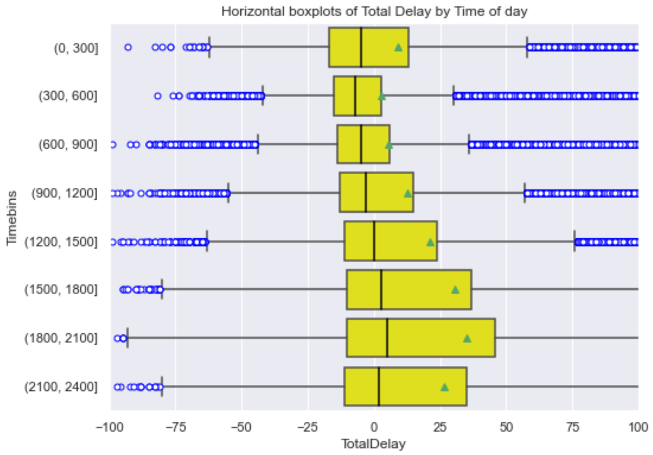
Before starting to answer the questions, the data has to be cleaned to make sure that the information is useful. Since the columns in the 2006 and 2007 year datasets were identical, they were combined to make a single dataset and was used throughout the analysis. The dataset was cleaned according to the requirements in each question. Unnecessary columns and missing values were removed when needed. There were no changes made to any of the data types of the variables. The scheduled departure times (CRSDepTime) was used where the time of flight was necessary. Delays are in minutes and a flight was considered to be delayed if delay was greater than 15. Delay equal to 15 means that the flight has departed/arrived on time and less than 15 means the plane is ahead of schedule.

WHEN IS THE BEST TIME OF DAY, DAY OF THE WEEK, AND TIME OF YEAR TO FLY TO MINIMISE DELAYS?

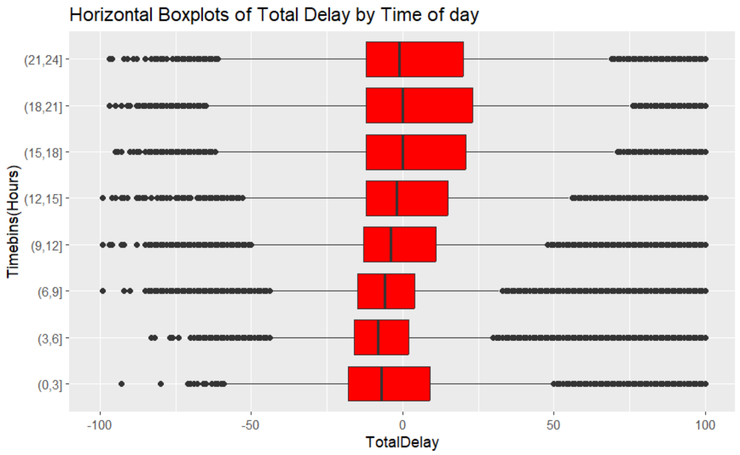
Missing rows in the departure and arrival time columns were excluded from the combined dataset. A new column was created to include total delay of the flight by adding departure delay and arrival delay. Then each part of the question was answered.

1. Best time of the day

For this part of the question, the time of day was split into 8 time bins with each bin having a width of 3 hours such as (00:00 - 03:00, 03:00 – 06:00, etc.). Then side-by-side boxplots were plotted for each time bin.



Side-by-side boxplots in Python

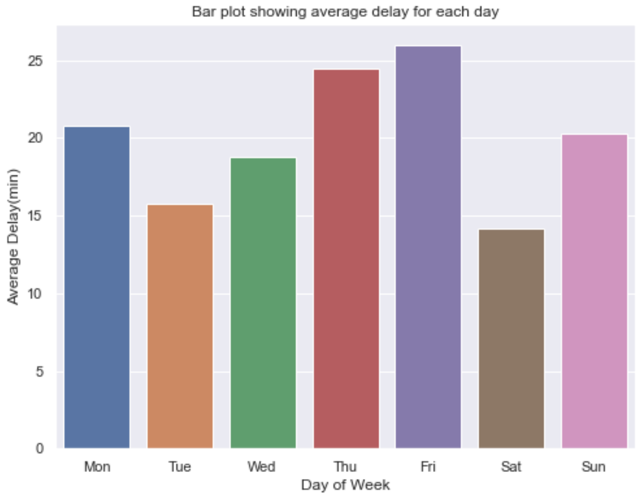


Side-by-side boxplots in R

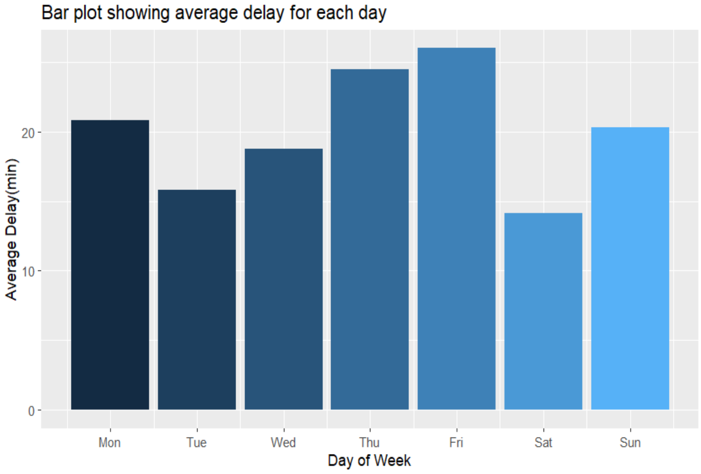
It is quite evident from the box plots that there are a large number of outliers. Due to this, the median is a better measure than the mean as the median does not get affected by outliers. The black vertical lines in each plot shows the median. The median of the time bin (03:00, 06:00) - 2nd time bin in Python and 7th time bin in R - has the lowest median. Therefore, the time bin (03:00, 06:00) is the best time of the day to fly to minimise delays.

2. Best day of the week

The dataset was grouped by the day of week and the average delays are displayed on a bar chart.



Bar plot in Python

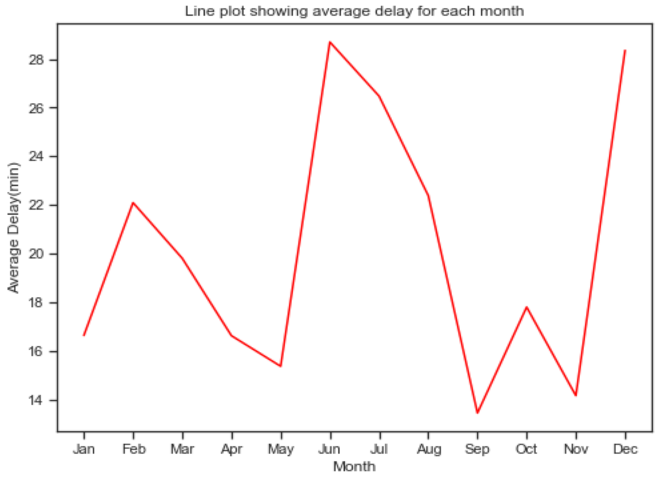


Bar plot in R

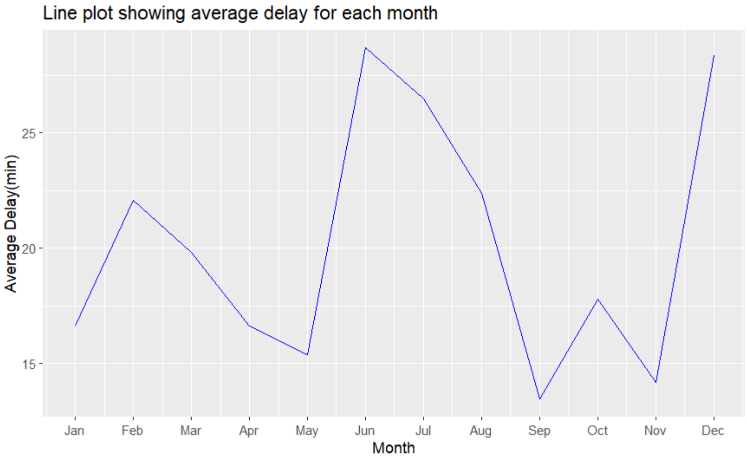
It is visible that Saturday has the lowest average delays out of all days closely followed by Tuesday. Therefore, Saturday is the best day of the week to fly to minimise delays. Thursday and Friday should be avoided as can be seen from the heights of the bar.

3. Best time of the year

Here, each month was taken as the time of the year to answer the question. Average delays were found for each month and a line plot was generated.



Line plot in Python



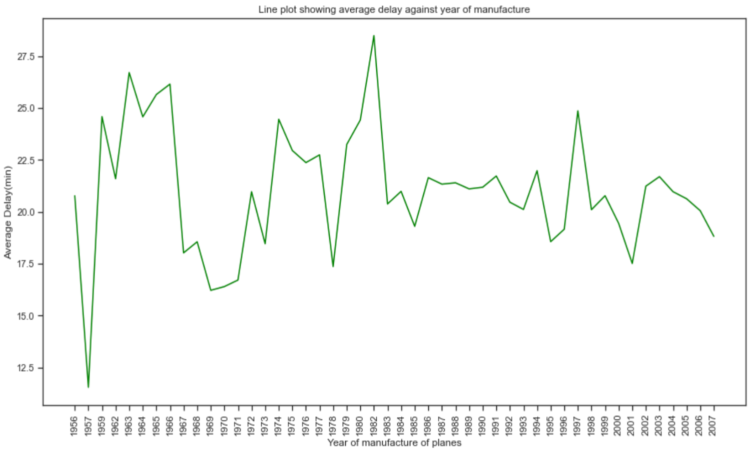
Line plot in R

From the above line plots, September and November has the least delays. We can conclude that September and November are the best months of the year to fly to minimise delays.

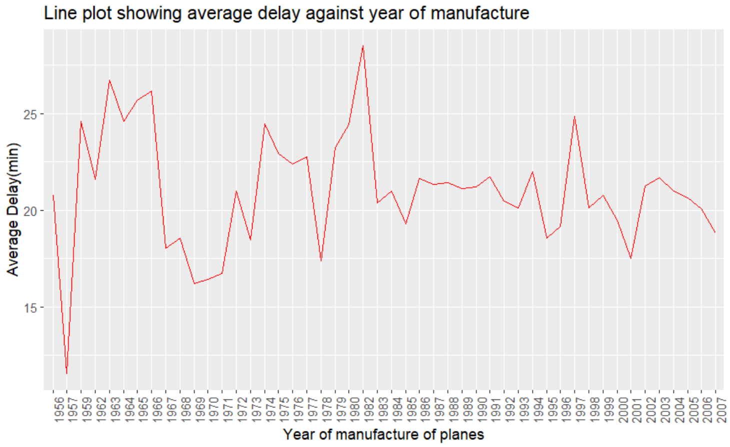
DO OLDER PLANES SUFFER MORE DELAYS?

Details of the planes were needed so the plane dataset was merged with the combined dataset. The manufacture year of the planes was associated with the age of the plane. Rows that did not contain departure times, arrival times and manufacture year were removed. While checking for unique values in year of manufacture, there were years such ‘0000’ and ‘None’. These were also removed from the dataset.

Then the dataset was grouped by year and the average delays are shown in a line graph.



Line plot in Python



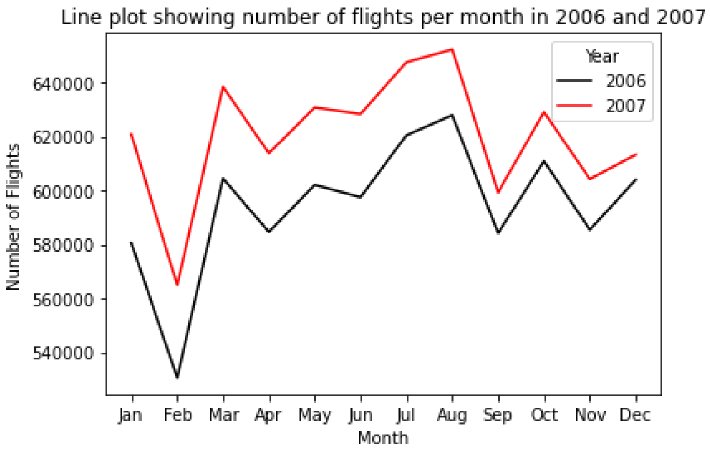
Line plot in R

There seems to be no particular trend in the line graph due to the many fluctuations. To see if there is any linear relationship between year of manufacture and average delay, the correlation coefficient is calculated.

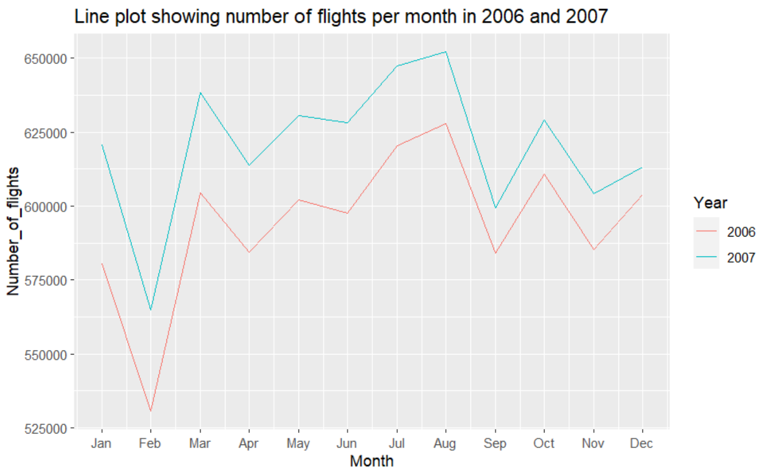
The value is -0.002 which is very close to zero. Both the line graph and the correlation coefficient lead to the conclusion that there is no relationship between plane manufacture year and average delays. We can say that older planes do not suffer more delays.

HOW DOES THE NUMBER OF PEOPLE FLYING BETWEEN DIFFERENT LOCATIONS CHANGE OVER TIME?

To answer this question, the dataset on airports is needed. Hence, it is imported and merged with combined dataset. Since there is no information on passengers, the number of flights is used instead. It is also assumed that all flights are full. The number of flights were found by grouping each month for both the years 2006 and 2007. To show the relationship between number of flights and months, two lines are plotted in the same graph.



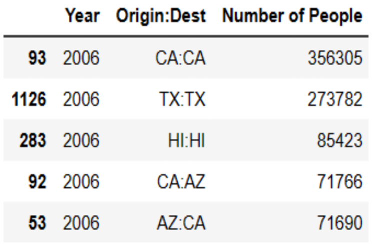
Line plot in Python



Line plot in R

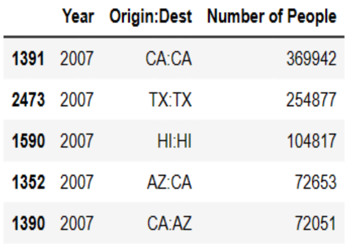
A key discovery is that there has been an increase in the number of flights in 2007 when compared to 2006. February seems to have the least flights while August is the busiest month in both the years. There is a steady rise in flights from March which peaks in August.

Further analysis was done to find the most used flight routes. States were taken as the origin and destination of the planes. The top 5 routes were found for each year and can be seen in the tables below.



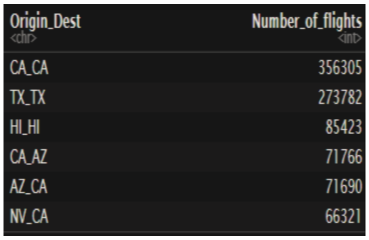
	Year	Origin:Dest	Number of People
93	2006	CA:CA	356305
1126	2006	TX:TX	273782
283	2006	HI:HI	85423
92	2006	CA:AZ	71766
53	2006	AZ:CA	71690

2006 table in Python



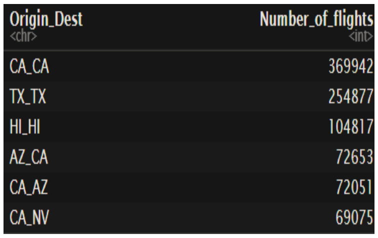
	Year	Origin:Dest	Number of People
1391	2007	CA:CA	369942
2473	2007	TX:TX	254877
1590	2007	HI:HI	104817
1352	2007	AZ:CA	72653
1390	2007	CA:AZ	72051

2007 table in Python



Origin_Dest <chr>	Number_of_flights <int>
CA_CA	356305
TX_TX	273782
HI_HI	85423
CA_AZ	71766
AZ_CA	71690
NV_CA	66321

2006 table in R



Origin_Dest <chr>	Number_of_flights <int>
CA_CA	369942
TX_TX	254877
HI_HI	104817
AZ_CA	72653
CA_AZ	72051
CA_NV	69075

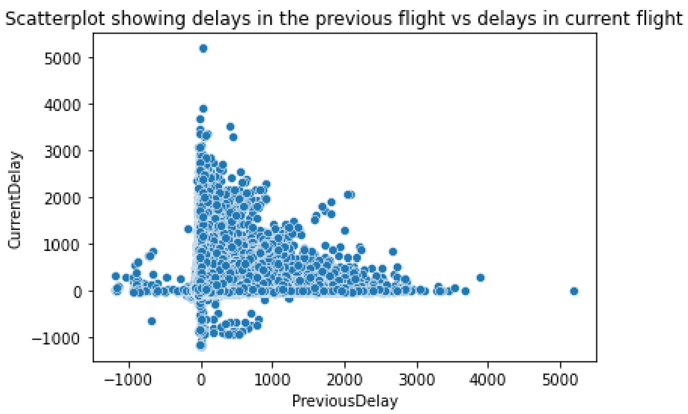
2007 table in R

In 2006, the top 5 routes with the greatest number of passengers travelling are CA:CA, TX:TX, HI:HI, CA:AZ, AZ:CA. This is identical in 2007 except AZ:CA and CA:AZ switching places.

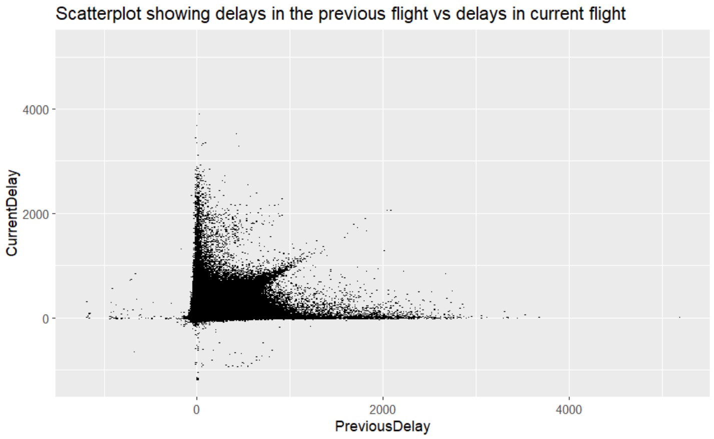
CAN YOU DETECT CASCADING FAILURES AS DELAYS IN ONE AIRPORT CREATE DELAYS IN OTHERS?

Rows that did not contain tail number, arrival delay and departure delay were removed. A datetime column was formed using the scheduled departure time of the flight and then the dataset was sorted by tail number and the datetime column to make it a continuous timeline. Afterwards, the delay of the same plane in the previous airport was found. Current delay is the total delay of the flight.

A scatter plot is plotted to show the association between delays in the previous airport and delays in the current airport.

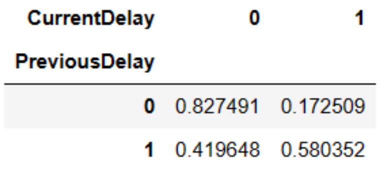


Scatter plot in Python



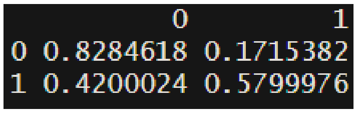
Scatter plot in R

It is not possible to clearly say that there is a relationship since the points are all over the place. To see whether cascading delays in one airport create delays in others, a crosstabulation with probabilities is formed. Since crosstabs work only for categorical variables, delays in previous and current airports were changed to 0s and 1s where 0 indicates there was no delay and 1 indicate there was a delay.



CurrentDelay	0	1
PreviousDelay		
0	0.827491	0.172509
1	0.419648	0.580352

Crosstab in Python



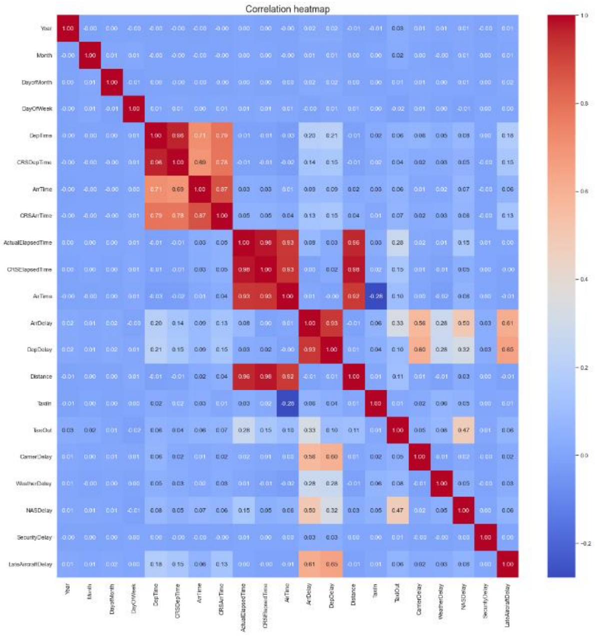
```
          0         1
0 0.8284618 0.1715382
1 0.4200024 0.5799976
```

Crosstab in R

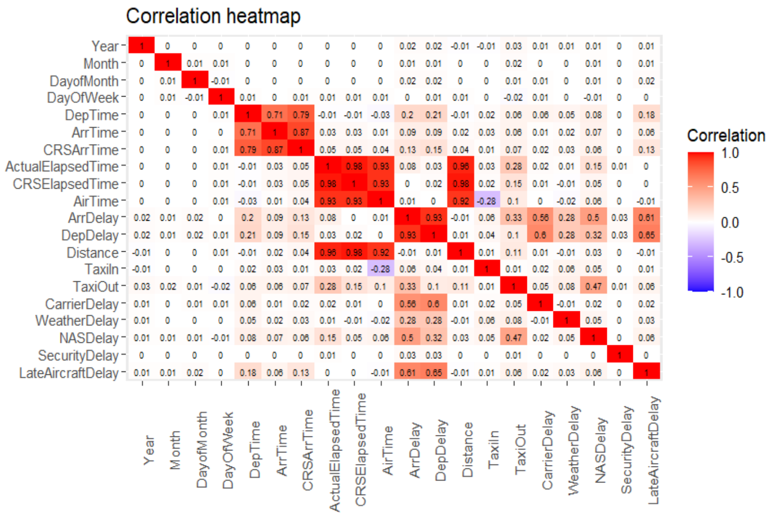
0.580 is the probability that there is a delay in the current airport given that there has been a delay in the previous airport which is greater than 0.420 which is the probability that there is no delay in the current airport given that there has been a delay in the previous airport. Therefore, the crosstab shows that cascading failures occur as delays in the previous airport create delays in the current airport.

USE THE AVAILABLE VARIABLES TO CONSTRUCT A MODEL THAT PREDICTS DELAYS

We predict that if there is a delay in the arrival of a plane or not. A Boolean column with 0s and 1s was created to show if there was a delay or not. For this, a logistic regression model is developed. The correlation between the available variables is explored in the heatmap.

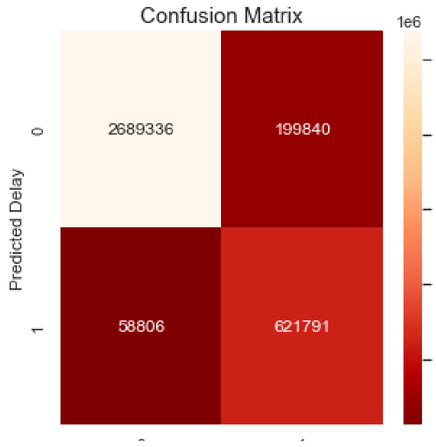


Correlation heatmap in Python

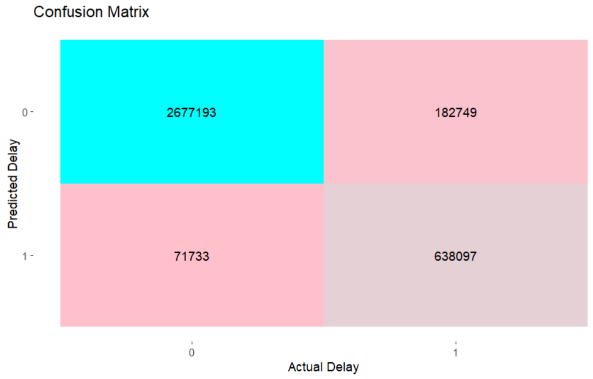


Correlation heatmap in R

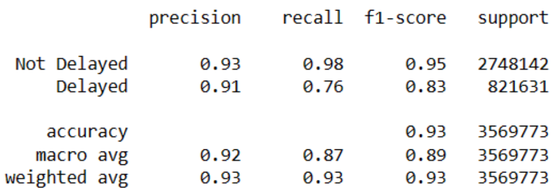
After having a look at the heatmap, the features variables were selected. As seen in the previous questions, the variables day of week, month, time had an impact on the delay. The departure delay and taxi out time along with the previously mentioned variables were chosen as the feature variables. The carrier, weather, NAS were not included in the model since the values will be unknown until the plane lands. After that, the model was developed and tested. A confusion matrix and model evaluation report are formed in both Python and R.



Confusion matrix in Python

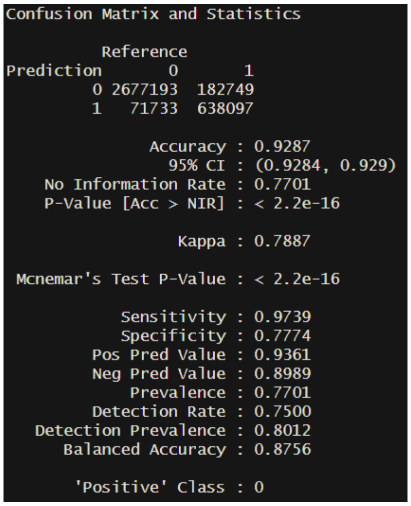


Confusion matrix in R



```
              precision    recall  f1-score   support

 Not Delayed       0.93      0.98      0.95   2748142
     Delayed       0.91      0.76      0.83    821631

    accuracy                           0.93   3569773
   macro avg       0.92      0.87      0.89   3569773
weighted avg       0.93      0.93      0.93   3569773
```

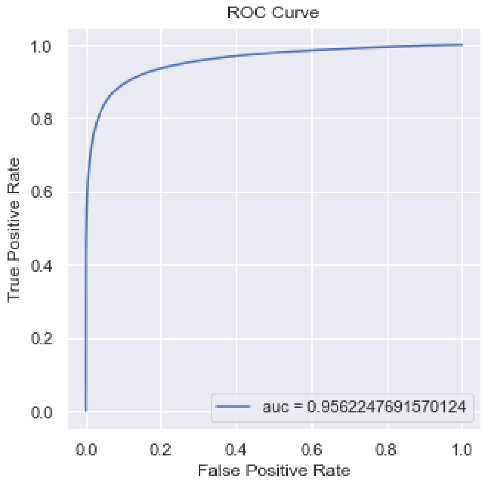
Classification report in Python



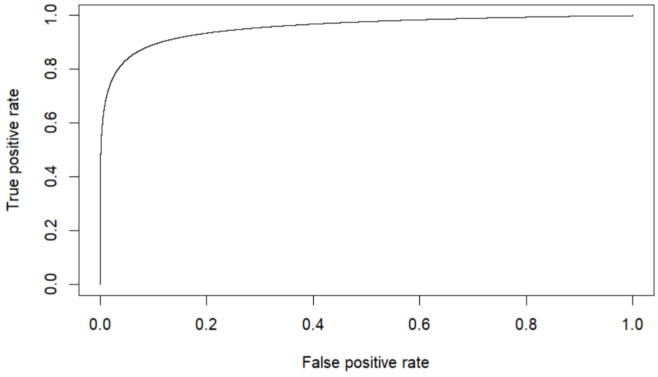
```
Confusion Matrix and Statistics

          Reference
Prediction       0       1
         0 2677193  182749
         1   71733  638097

               Accuracy : 0.9287
                 95% CI : (0.9284, 0.929)
    No Information Rate : 0.7701
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7887

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9739
            Specificity : 0.7774
         Pos Pred Value : 0.9361
         Neg Pred Value : 0.8989
             Prevalence : 0.7701
         Detection Rate : 0.7500
   Detection Prevalence : 0.8012
      Balanced Accuracy : 0.8756

       'Positive' Class : 0
```

Model evaluation report in R

Accuracy was not used as a metric to identify the model suitability because there is a class imbalance. In the report generated in Python, the f1-score is very high (0.95 for not delayed and 0.83 for delayed) which is very good. In R, the sensitivity (0.97) and specificity (0.78) are acceptable. These values give a good indication that the models built in Python and R are appropriate. Finally, ROC curves are plotted and the area under the curve is calculated.



ROC curve in Python



ROC curve in R

The area under the curve (auc) values calculated in Python (0.956) and R(0.956) are close to 1 which confirms our previous indication that the logistic regression model constructed is suitable to predict arrival delays.