Analysis of Data Expo 2009: Airline on time data

By Yuwanath De Silva

Table of Contents

Executive Summary

The two years chosen for the purpose of answering all the questions were 2006 and 2007 with a focus on departure delays. The choice of years was due to the closest proximity to the present. Departure delay was focused on judgment thought from a passenger's perspective who would be a lot more interested in knowing whether or not their flight is going to depart late at the point of booking. Also, it was intuitive at the start to think that arrival delays without departure delays are uncommon and if anything, it was common for arrival delays to be caused due to departure delays.

Each question was done on separate Jupyter notebooks and R scripts to compensate for memory/computing constraints importing only the relevant columns necessary to answer each question. Please find the code in files named Q1-Q5.jpynb and Q1-Q5.R in Jupyter notebooks and R scripts, respectively. Note that 2006.csv, 2007.csv and plane-data.csv files should be present in the directory from where the code is run.

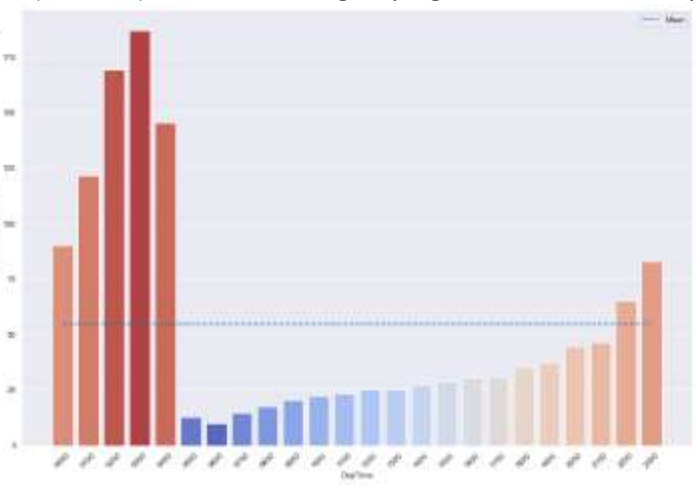
Figures given alongside the analysis were obtained using python, and the method explained is similar for both R and python. Identical figures obtained using R are given at the end of the report.

Q1. When is the best time of day, day of the week, and time of year to fly to minimise delays?

The process of answering this question and all others starts with importing the necessary libraries which allow the relevant datasets to be transformed for analysis. For this question, NumPy, pandas, matplotlib, and seaborn from python and data.table , tidyverse, lubridate, dplyr, and ggplot2 from R were imported. All of which were used for the purposes of data manipulation and data visualization.

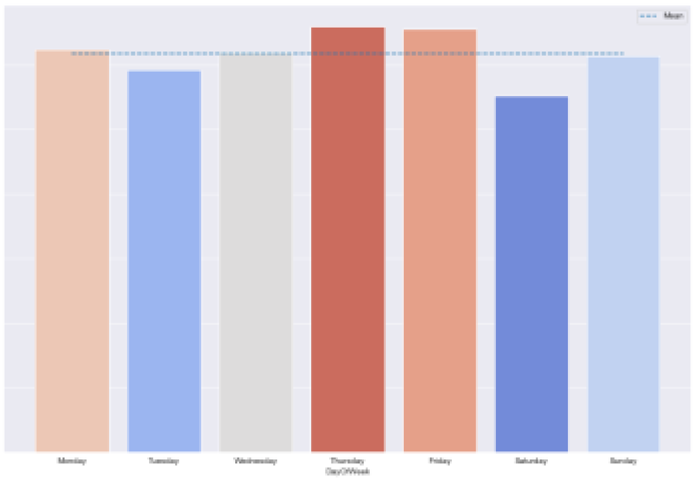
The concatenated data was cleaned to account for null values present and filtered to records of only delayed flights as that is what the question demands. The Departure time column was prepared for the grouping by taking the last two digits (minutes) out since the grouping was done at hourly intervals.

Figure 1



After which the mean value of departure delay was grouped hourly. The mean was taken to account for the uneven number of flights departing each hour which will be factored in if summed up. The choice of plot for the visualization was a bar plot with a diverging colour palette to clearly communicate which hours to be avoided when flying. The average line is also plotted for reference. Red bars indicate a period from 2200-0400 when delays seem to be the worst, with flights departing between 0300-0400 clocking the worst mean delay of 187 minutes. From 0500 till 2100 planes seem to suffer relatively low delays with the best time slot to fly proving to be between 0600-0700.

Figure 2



Similarly, for the second part of question 1, the mean value of departure delay was grouped by the day of the week. Here the coding given in the dataset [DayOfWeek 1 (Monday) - 7 (Sunday)] was replaced for a more convenient interpretation. The bar plot with the diverging colour pallet again communicates the inferences at a glance. The dotted line shows the average delay among the days at 31 minutes. The darkest red bar suggests that Thursday suffers the most delays, while Saturday, the darkest blue bar suffers the least, proving to be the best day of the week to fly to minimise delays.

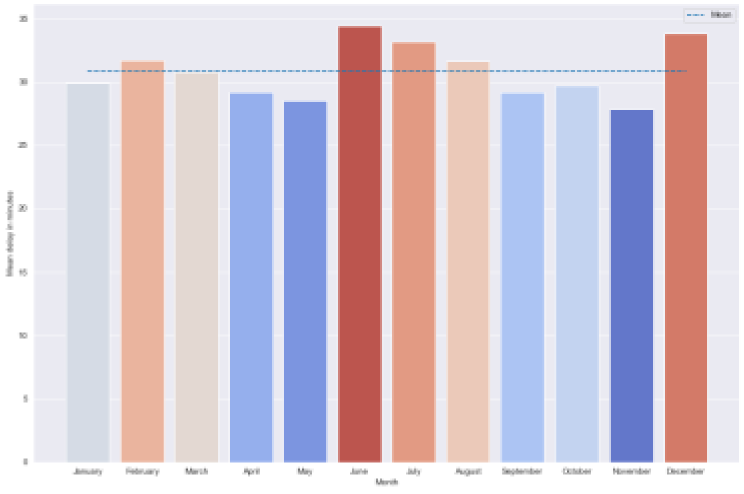


Figure 3

Keeping in line with the same approach as the last two parts, the query for the third part ran a group by to capture mean delays amongst the twelve months. The data given in numeric format for months was transformed into months in words for convenience. As evident by the plot, June seems the month to avoid while November is the best time of the year to fly to minimise delays since it has the least average delay of 28 minutes.

Overall, combining the efforts of all three parts we can narrow down the optimum moment in time to take a flight to be in November on a Saturday between 0600-0700. The data suggests minimum delays for that very particular slot.

Q2. Do older planes suffer more delays?

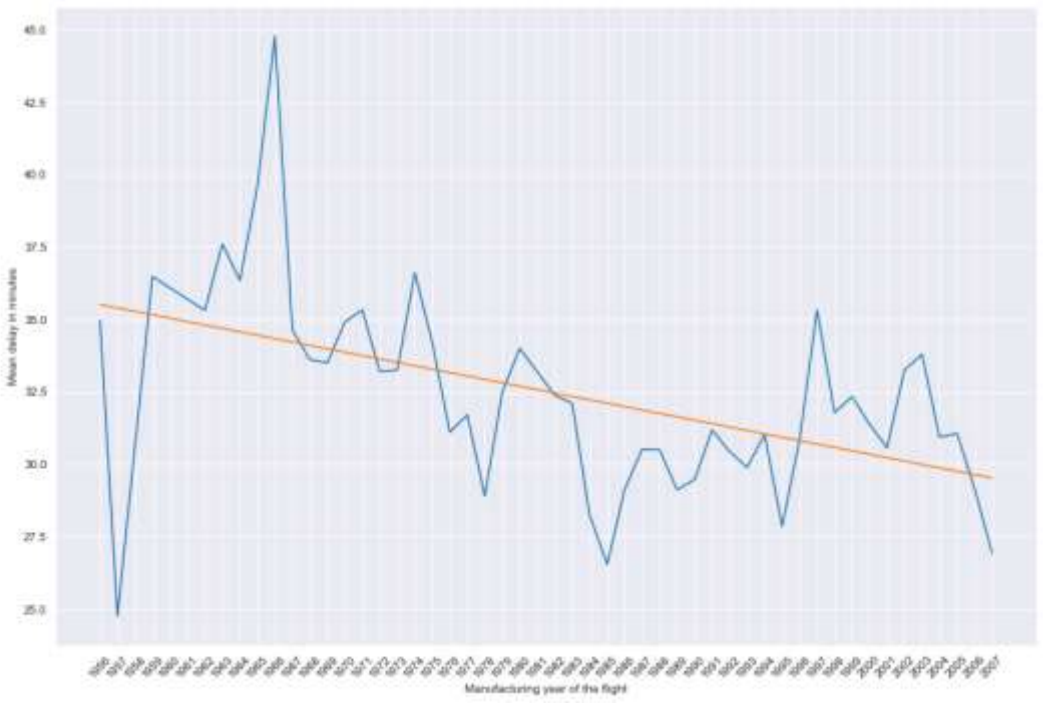
Now that we know when delays occur it will be helpful to hypothesize why those delays occur in the first place. One such hypothesis could be that the aircraft deployed for the flight has a part to play in the delays caused. More specifically the age of the aircraft may be a determinant of the delays. In this question, we try to examine the validity of that hypothesis.

The same libraries used for the previous question were used for this one. One extra dataset containing data about the airplanes was used in addition to the previous 2006 and 2007 datasets. The flight data was filtered to get the records with delays only, while all datasets were cleared of null values and some incompatible values in the manufacturing year column. The working dataset was created using an inner mage on tail number, the unique identifier of each aircraft. This data was then grouped by manufacturing year of the aircraft and the mean value of departure delay associated.

The line plot produced clearly shows a declining trend, as also indicated by the trendline. Planes with an early manufacturing year seem to suffer greater delays, supporting the hypothesis that older planes suffer more delays. The fleet used between 2006-2007 has had planes with ages ranging from 01 to 51 years. With access to better technology in aircraft engineering and less wear and tear it is intuitive why younger planes may suffer less delays while planes which have been in service for a longer period of time and the latter part of their lifecycle are more prone to technical failures, hence greater delays.

Passengers can use this data to track down the manufacturing year of the airplane of their flights to make a better-informed decision at reservation while airlines and airports are also able to make better calls when adding new airplanes to their respective fleets.

Figure 4



Q3. How does the number of people flying between different locations change over time?

As humanity evolved so did the modes of transportation. The reasons for travel have become more complex and the developments in the aviation industry have allowed for greater accessibility for a wider range of customers and hence greater mobility to our civilization.

In this question, the count of flights was used as a proxy for the number of people flying due to the unavailability of that data explicitly. Since each flight carries about 100 passengers on average, subject to the size of the aircraft, the real volume would be ten-fold the values calculated using the proxy.

After the data was cleaned the two years were plotted using separate lines in the same graph for comparison. From that, it became evident that the later year, 2007, saw an increase in volume compared to 2006. The 2007 line stays well above the 2006 line with an average difference of 25941 flights more, albeit the similarity in the pattern followed over the months which suggests seasonality.

Further, when the count of flights was grouped on origins 185 out of 285 airports saw an increase in flight counts in 2007 compared to 2006, which is almost 65% of locations showing an increase in the number of people flying. Although cluttering to visualize the change among all 285 locations the numbers give a good indication of how the number of people flying between different locations has increased over time.

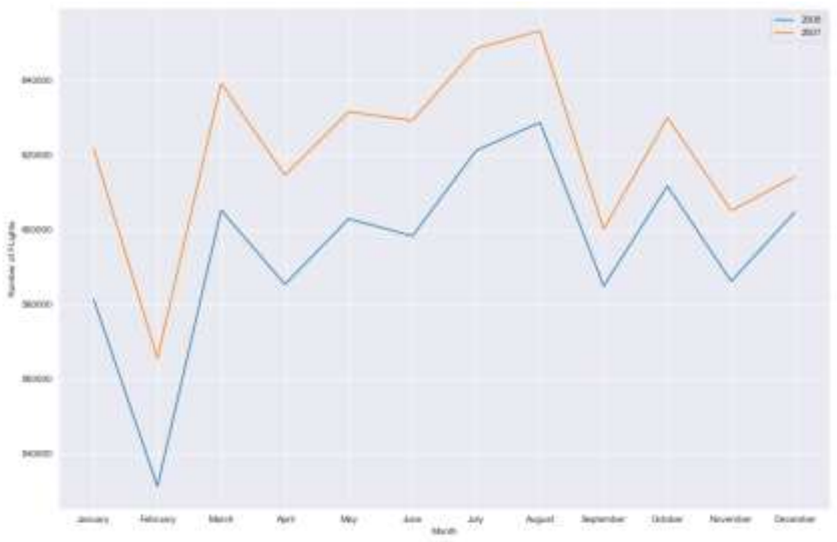
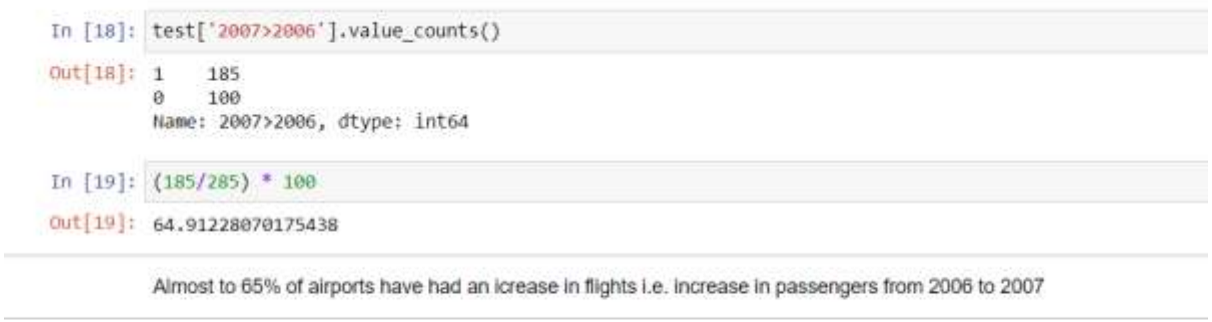


Figure 5



```
In [18]: test['2007>2006'].value_counts()

Out[18]: 1    185
         0    100
         Name: 2007>2006, dtype: int64

In [19]: (185/285) * 100

Out[19]: 64.91228070175438
```

Almost to 65% of airports have had an icrease in flights i.e. increase in passengers from 2006 to 2007

Figure 6

Q4. Can you detect cascading failures as delays in one airport create delays in others?

Examining the proceeding effects of a delay at one airport helps us understand whether or not an aircraft is able to return back to its schedule. However, recovering from a delay is not the easiest of tasks and one delay can have a cascading effect down the aircraft's planned schedule. In this query, we attempt to detect such cascading delays.

In addition to the set of libraries used in the previous question, datetime was used for the purpose of creating a timestamp for each of the records which would allow us to track an aircraft down its path in the correct order. In order to use seaborn with R, Reticulate library was used which embeds a Python session to R. Subsequent to the cleaning of data, and the creation of the timestamp column which was used to sort all the records, a function was run to group and unstack the dataframe in such a way that the index of the column was the tail number i.e. a unique aircraft and the columns indicated its departure delay values from the initial dataset in the same order as their occurrence.

This transformation allowed for a 2D dataframe well suited to be visualized by a heatmap. Due to the size of the dataset i.e. the number of airplanes recorded, it was decided to visualize 100 airplanes at a time with 100 of their records. This captures a snapshot of an aircraft's journey with its delay values.

The diverging colour palette is centered at 0 and maxed at 60 to communicate all delays in red with the intensity increasing as the delays become greater. Delays greater than 60 minutes (an hour) are shown with the highest intensity of red and can be seen perpetuating for a longer period of time. On time and early (minus values) departures are depicted using the blues.

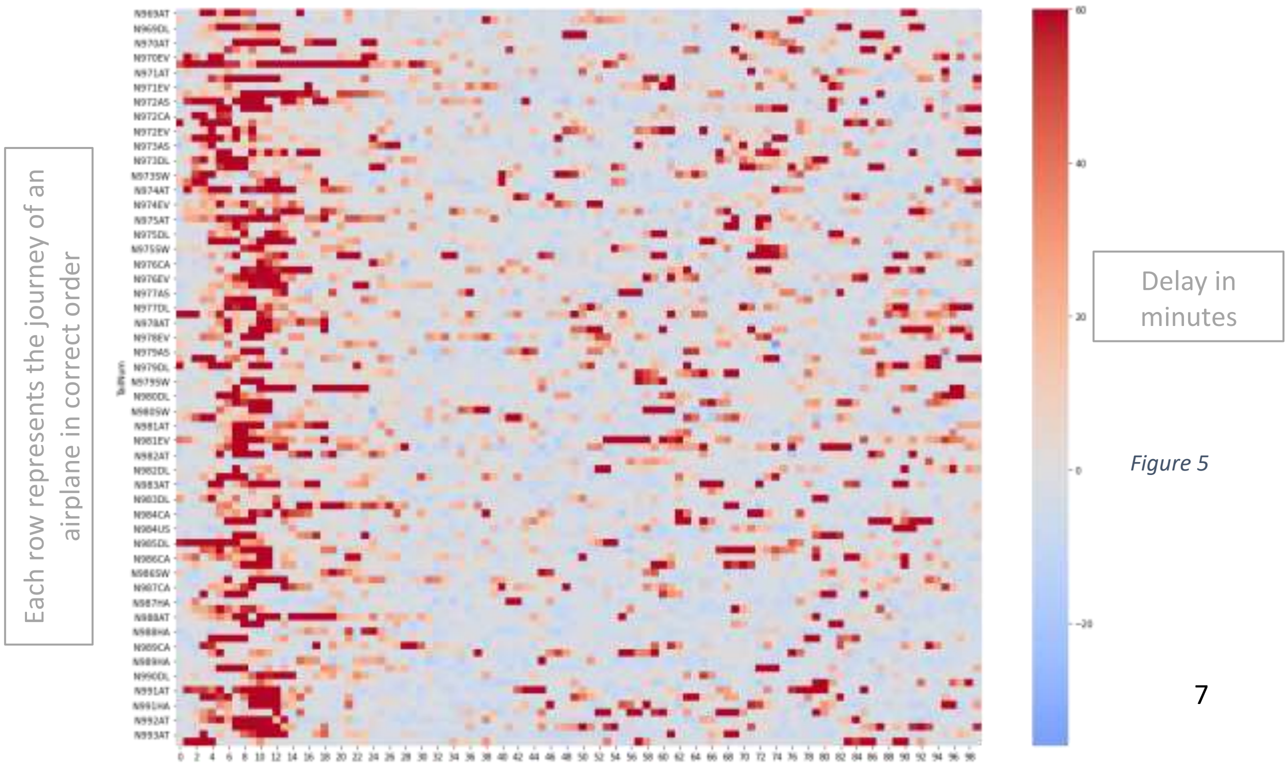
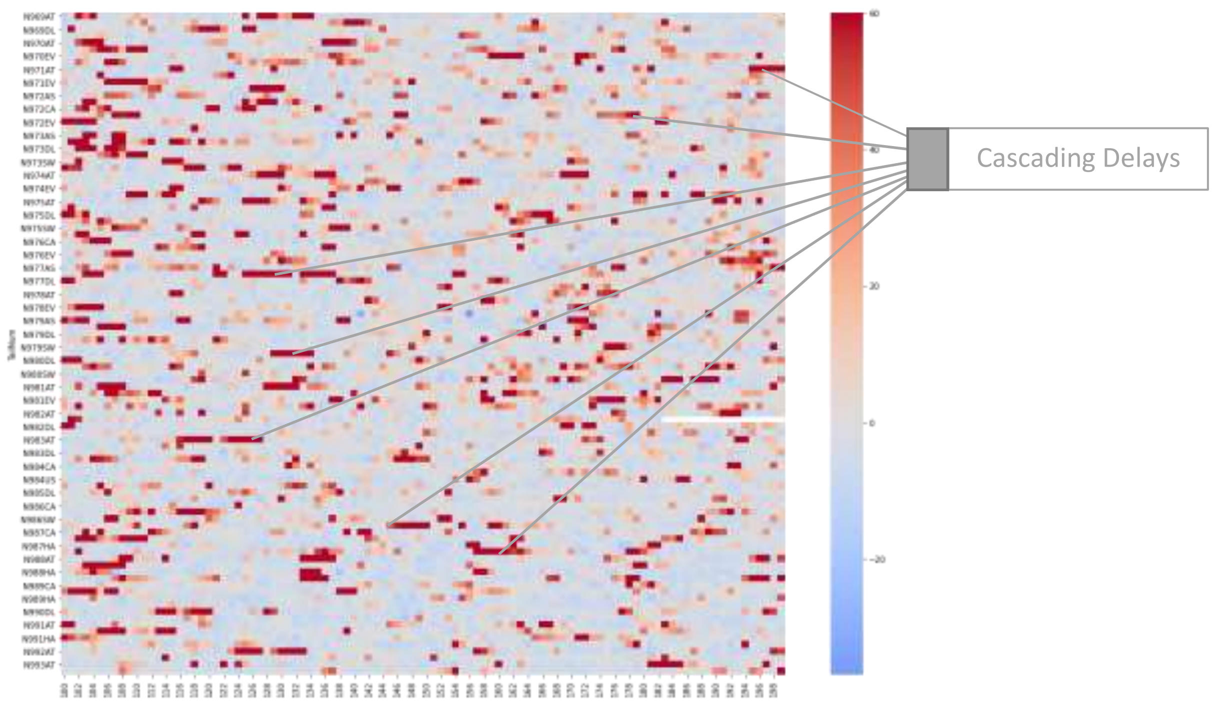


Figure 5



The above is the next snapshot for the same set of airplanes i.e. the delay values for their next hundred records. The code can be altered to give a snapshot of the required aircraft for a required period of time, increasing its usability.

The horizontal clusters of red suggest the cascading delays of airplanes, an increase of redness in those same clusters shows the worsening of the delay while return to normalcy is shown by the shading out effect.

As the detections are scanned, it seems that the cascading effect present dies down after a few destinations. It is reasonable to assume that an aircraft is able to break free of the perpetual delays when given a long layover or a long break before embarking on the next flight.

Airplanes which have minus delay times, also seem to carry the effect forward although not as intensely.

Cascading delays can cause detrimental effects on an airline. The first step of dealing with them is to detect which airplanes seem to suffer from them. Preventive methods can then be prescribed to alter the pattern. For instance, even if a delay does occur at a certain point the airline can try to compensate for it at the next stop by being better prepared at the receiving airport preventing the lag from perpetuating. After implementing any such measure, the new snapshots can be compared with the previous ones to check for efficiency.

Q5. Use the available variables to construct a model that predicts delays.

One of the most worthwhile use cases of the millions of data present is to use it to predict delays, so that the decision-makers can be better prepared. Machine learning, classification in particular was the tool of choice to tackle this question. Classification is a predictive model which can detect whether or not a delay will occur given the training from the input data. Five different classification models, namely Random Forest, Decision Trees, Logistic Regression, K-Nearest Neighbors and MTP neural networks were trained and evaluated with the Random Forest model proving to be the best fit.

Apart from the previous set of libraries, sklearn library was used to import many tools required to construct the model.

The features used for the model were “Year”, “Month”, “DayOfWeek”, “CRSDepTime”, “CRSElapsedTime”, “FlightNum”, “TailNum”, “UniqueCarrier”, “Distance”, “LateAircraftDelay” and “ManuYear”. Since they came from three different datasets, they were concatenated and bound appropriately after which the final dataset was cleaned.

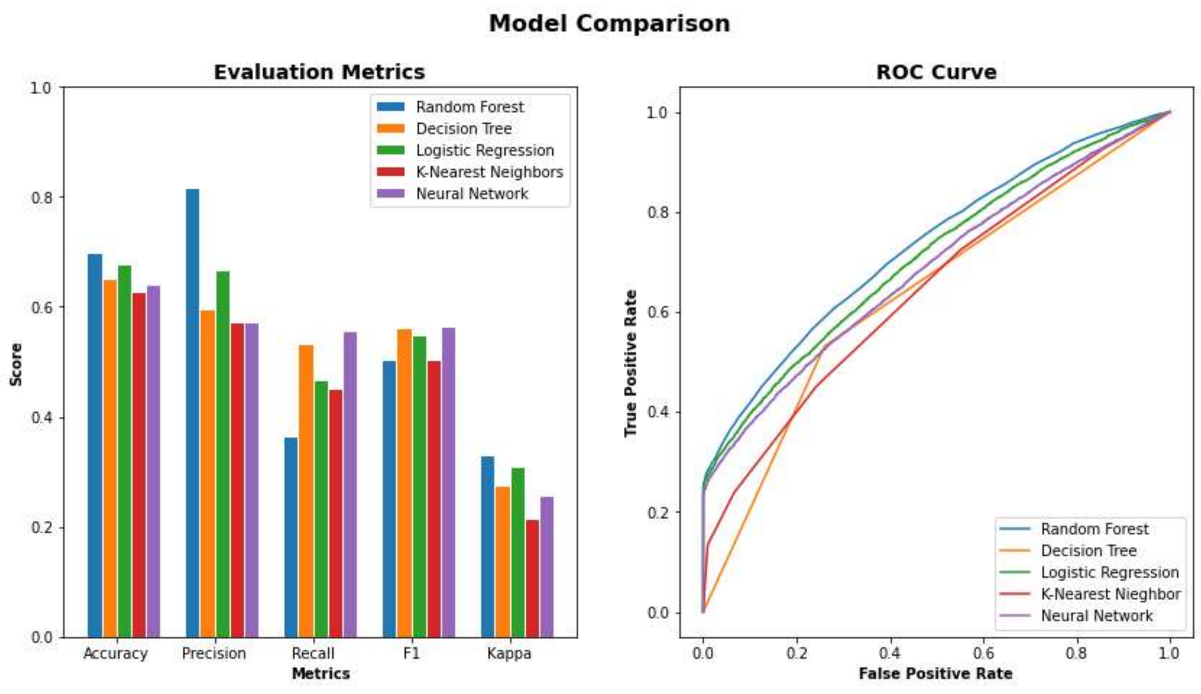
Due to computational/memory constraints, the final dataset was a representative sample of 65000 rows with a 70%/30% split for training and testing, respectively.

The numeric features used, "Distance", "CRSElapsedTime" and "LateAircraftDelay" were scaled using the standard scaler to prevent distortion given the differences in the range of the values. All three of these features will be known prior to a delay occurring which makes them suitable to be included in the model while most other numeric features present in the dataset would be known only after the fact. Note that late aircraft delay is a feature which indicates arrival delay at an airport due to the late arrival of the same aircraft at a previous airport and CRS data are specified at the point of obtaining the certification for release to service.

The categorical features used, “Year”, “Month”, “DayOfWeek”, “CRSDepTime”, “FlightNum”, “TailNum”, “UniqueCarrier” and “ManuYear” were encoded using one hot encoding to translate the data into what the computer could understand; 1s and 0s. These particular features were also carefully chosen from the features available in the datasets. The features indicative of time made the cut due to the seasonality of the delays which can help train the model for similar timeframes. “CRSDepTime” was pre-processed into hourly chunks to avoid making the training set a lot larger. “FlightNum”, “TailNum”, “UniqueCarrier” and “ManuYear” are all features from which the model can learn about the aircraft which can help its prediction given the same planes will be in service for a period of time.

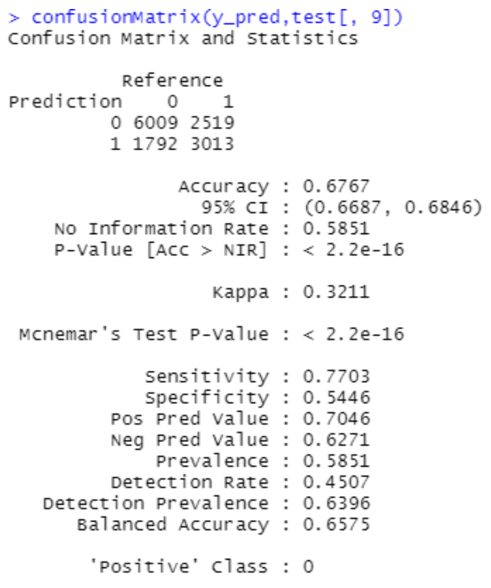
The target feature 'DepDelay" was also encoded with 1s and 0s; one for a delay and zero for not.

For evaluation purposes, a function was introduced to keep iterating over the five models. Accuracy, Precision, Recall, F1 Score, Cohens Kappa Score, Area Under Curve and Confusion matrix were the measures used for evaluation. A comparative visualization is shown below to summarise the performances of the five models.



With an accuracy of almost 70%, precision of 82% and an AUC value of 0.73 the random forest model was the best performing model out of the five that can be further improved subject to tuning and better-quality data. Provided with more computational power the model could have also been trained with a larger amount of training data which also has the potential to improve performance.

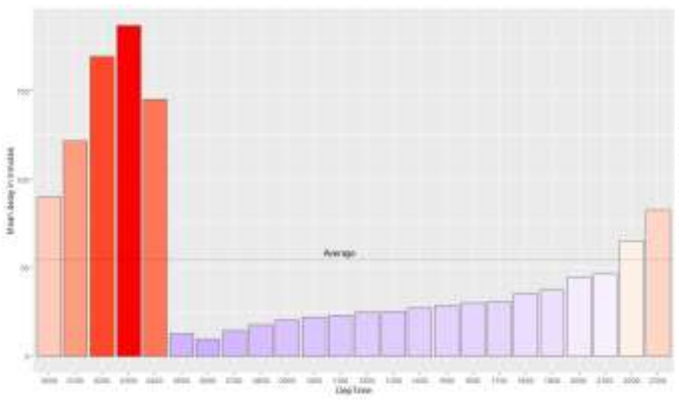
Computational constraints were faced much more in R, hence only one model could be trained with an even smaller sample size. The libraries used in R were caret and randomForest. The as factor() built-in R function allowed to encode categorical features. Even with a smaller sample size and limited features the random forest model deployed was able to reach an accuracy level of 67% and a Kappa Score of 0.32.



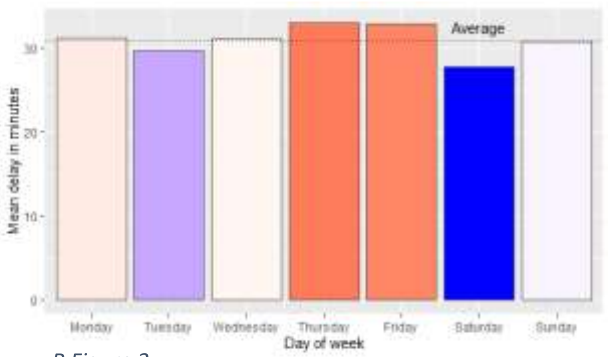
```
> confusionMatrix(y_pred,test[, 9])
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6009 2519
         1 1792 3013
                                          
               Accuracy : 0.6767          
                 95% CI : (0.6687, 0.6846)
    No Information Rate : 0.5851          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.3211          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7703          
            Specificity : 0.5446          
         Pos Pred Value : 0.7046          
         Neg Pred Value : 0.6271          
             Prevalence : 0.5851          
         Detection Rate : 0.4507          
   Detection Prevalence : 0.6396          
      Balanced Accuracy : 0.6575          
                                          
       'Positive' Class : 0               
```

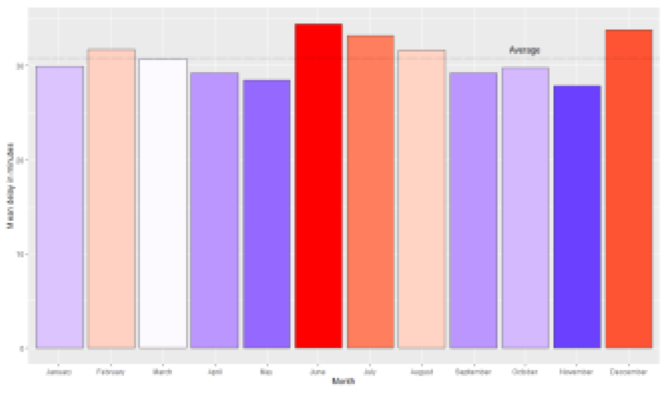
R Plots



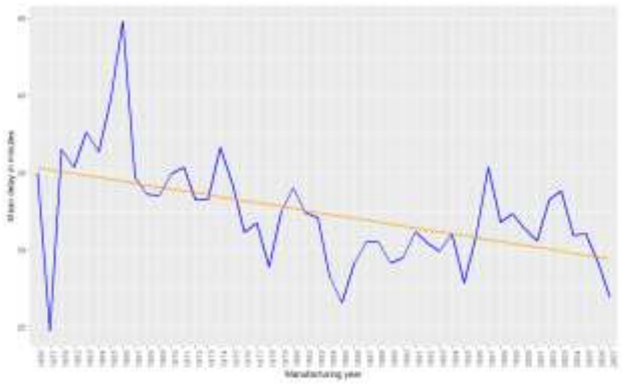
R Figure 1



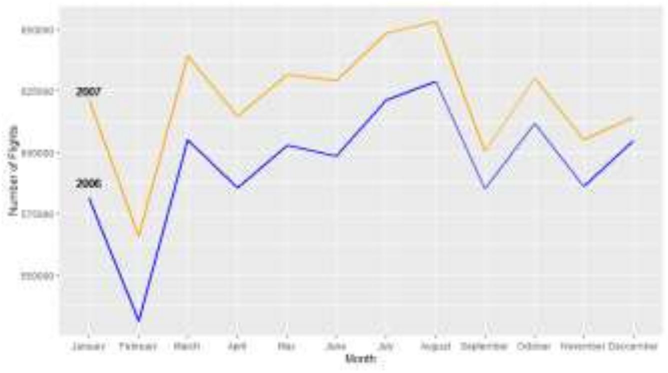
R Figure 2



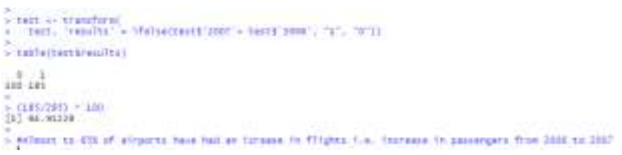
R Figure 3



R Figure 4



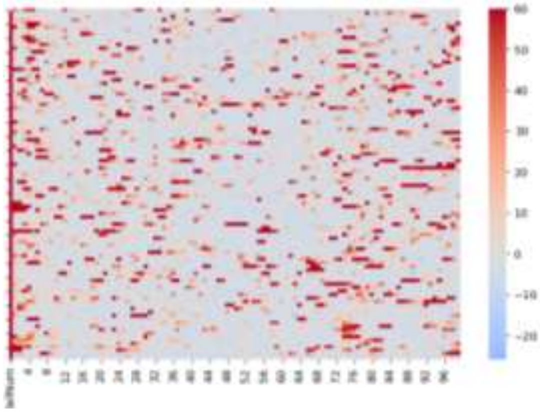
R Figure 5



```
>
> test <- transform(
+   test, 'results' = ifelse(test$'2007' > test$'2006', "1", "0"))
>
> table(test$results)

  0   1
100 185
>
> (185/285) * 100
[1] 64.91228
>
> #Almost to 65% of airports have had an increase in flights i.e. increase in passengers from 2006 to 2007
```

R Figure 6



R Figure 7

References

Harvard Dataverse. (2008, October 6). *Data Expo 2009: Airline on time data* (Version 1.0) [Dataset]. Harvard Dataverse. https://doi.org/10.7910/DVN/HG7NV7