ST2195 Programming for Data Science

Coursework Project

UOL Student ID: 200645887
Module: ST2195 - Programming for Data Science
Page Count (Excluding table of contents and bibliography): 10 pages

Table of Contents

Introduction

This report aims to deliver an analysis on flight details for all commercial flights on major carriers within the USA for the years 2006 and 2007. This data is a subset of the “2009 ASA Statistical Computing and Graphics Data Expo” which consists of such flight from October 1987 to April 2008. The complete dataset along with its supplementary datasets and related information are hosted on the Harvard Dataverse.
Any and all stakeholders of domestic US commercial flights can expect to gain more insight into flight delays within the USA, through the analysis delivered in this report. However, it is important to take note of the limitations of this analysis, as only a re-creatable random 50% sample of the 2 years’ worth of data was taken to conduct the analysis. And the 2 years under consideration do not reflect well in terms of recency. This report will detail the many steps taken on this little journey of data science, to deliver this analysis to you.

Data cleaning and preprocessing

The 2 datasets for the years 2006 and 2007 combined had 14,595,137 entries and 29 columns between them. Missing values were checked for and it was observed that more than 14,000,000 of the entries of the cancellation column were missing and so the column was dropped as it was virtually non-existent. Duplicate entries were dealt with in the same manner.

Many types of delay related values were present in the dataset. However, it was decided to mainly be concerned with arrival delay when it comes to tackling any delay related question. This is due to the assumption that passengers and airline operators are more likely to be concerned with the impact of their flights arriving late at the intended destination rather than the impact of the flight departing late from its origin location. This assumption was made ahead of any data manipulation in order to be able to carry out the analysis and modelling in a concrete manner.It was observed that cancelled and diverted flights do not have any arrival delay values recorded and thus all such flights were dropped form the dataset. As this resulted in their being no cancelled and diverted flights in the dataset it was decided to remove those columns entirely.

When checking for the maximum and minimum numeric values of the dataset it was revealed that some of the time values encoded in the 24 hour time format exceed the 24 hour time limit. Along with this it was also revealed that the scheduled elapsed time and air time columns had taken on some negative values, which is not possible as both columns record only elapsed time values and do not record any time saved values. Columns which had 24 hour time format values which were recorded as regular integers; were converted to date time objects in Python which can only contain values that represent an actual time in 24 hour time format. This was done in order to get rid of any invalid time values and drop them as missing values. As an example since integers have no restrictions time values such as 1161 which would be read as 11:61 pm could also exist, which is an invalid time and should be dropped from the data. The negatively elapsed time values were dealt with by only keeping positive values in those columns.

The ‘Time_bin’ column was created to answer question 1, by binning the hours of the scheduled departure time column into 12 time slots, of 2 hours each. The year column of the plane dataset was renamed as the ‘Manufactured_year’ column and it was observed that there were a few implausible year values in that column such as ‘0000’, ‘None’ and ‘2008’ as our dataset focuses only on the years 2006 and 2007, thus these values were removed. Afterwards the plane dataset was merged with the main dataset to create a new column. The new column named ‘Age’ was created to answer question 2, this was created by subtracting the Manufactured year of a plane from its currently flying year. There seemed to be planes of negative 1 year of age in the dataset and those entries were removed. The ‘Year_month’ column was created to answer question 3. This was created by merging a flight’s corresponding year and month together into one value of the format “year-month”. The ‘Date_time’ column was created to answer question 4. This column was created by merging a flight’s respective year, month, day of month and time into one value, in the format of “year-month-date-time”. A new column ‘Delay_status’ was created for question 5. The purpose of this column is to indicate whether a particular flight had experienced an arrival delay or not. Finally, due to computing power limitations and time constraints a reproduceable random sample of 50% of the dataset’s entries was generated and used as the final dataset to conduct all analysis and modelling.

Question 1: When is the best time of day, day of the week, and time of year to fly to minimise delays?

Best time of day

The flights were grouped by the time slots in the ‘Time_bin’ column and the average delay and maximum delay within 1 standard deviation above the mean was calculated to find the time slot when passengers are most likely to minimze their arrival delay.

Colour-blind friendly colour palettes were used for all visualizations in both python and R.

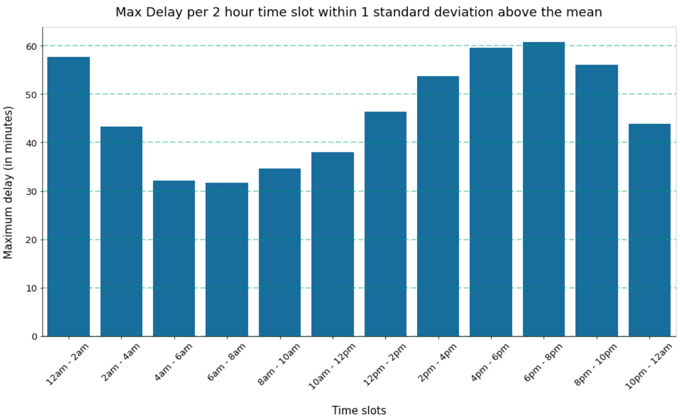


Figure 1:Max delay per 2 hours in Python

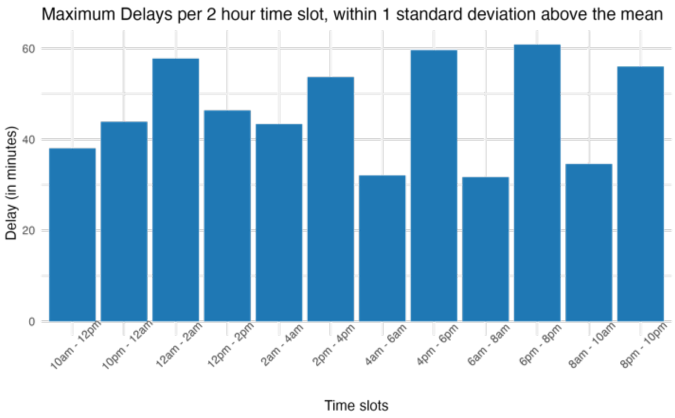


Figure 2:Max delay per 2 hours in R

Time Slot	Average Delay	Maximum Delay (1 standard deviation above the mean)	Number of flights
6am – 8am	1.362288	31.706949	795718
4am – 6am	0.741603	32.083396	42725

Since the maximum delay within 1 standard deviation above the mean for the 6am - 8am time slot is the lowest maximum delay from all the time slots, it can be said that between 6am - 8am is when passengers are most likely to minimze their arrival delay.

Best day of the week

To identify the best day of week, the flights were grouped by the day of the week and the average delay and maximum delay within 1 standard deviation above the mean was calculated to find the day of the week when passengers are most likely to minimize their arrival delay. The days are encoded as 1 – Monday through 7 – Sunday.

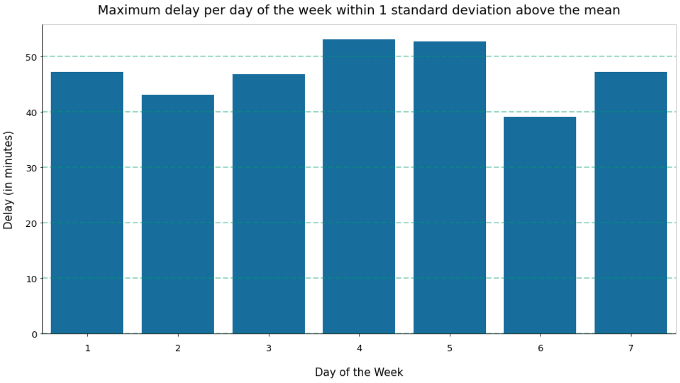


Figure 3:Max delay per day in Python

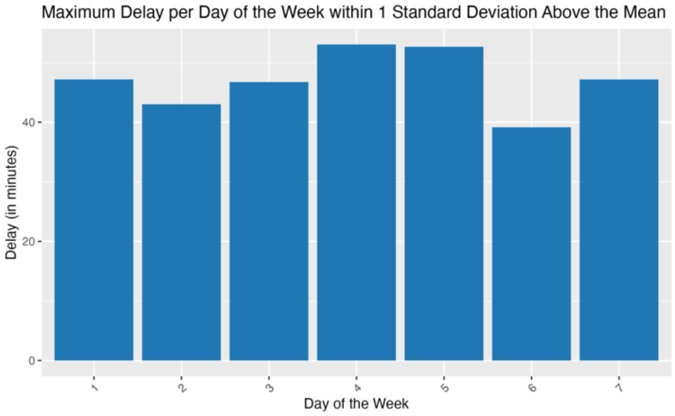


Figure 4:Max delay per day in R

Day of Week	Average Delay	Maximum Delay (1 standard deviation above the mean)	Number of flights
Saturday	5.491551	39.142201	727147
Tuesday	7.462359	43.019327	827799

Since the maximum delay within 1 standard deviation above the mean for Saturday is the lowest maximum delay from all the time slots, it can be said that Saturday is when passengers are most likely to minimze their arrival delay.

Best time of year

To identify the best time of year, the flights were grouped by the months and the average delay and maximum delay within 1 standard deviation above the mean was calculated to find the month when passengers are most likely to minimize their arrival delay. The months are encoded as 1 – January through 12 – December.

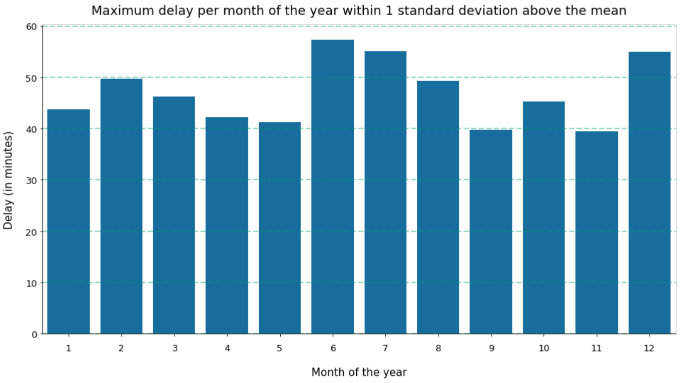


Figure 5:Max delay per month in Python

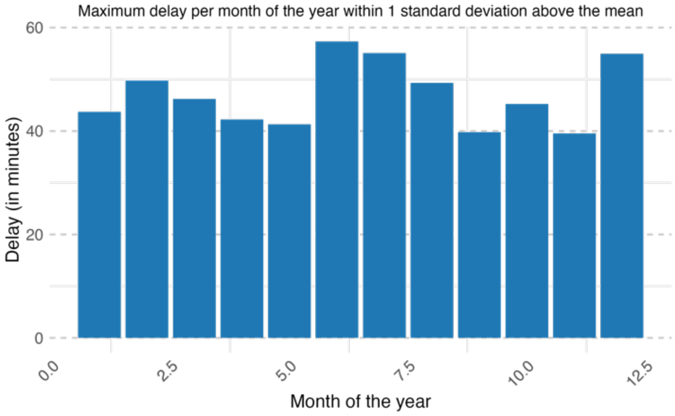


Figure 6:Max delay per month in R

Month	Average Delay	Maximum Delay (1 standard deviation above the mean)	Number of flights
November	6.042599	39.526019	483131
September	5.941374	39.772763	476497

Since the maximum delay within 1 standard deviation above the mean for November and September are the lowest maximum delay from all the months it was decided to encode the months to seasons to check which season is the best time of year to fly.

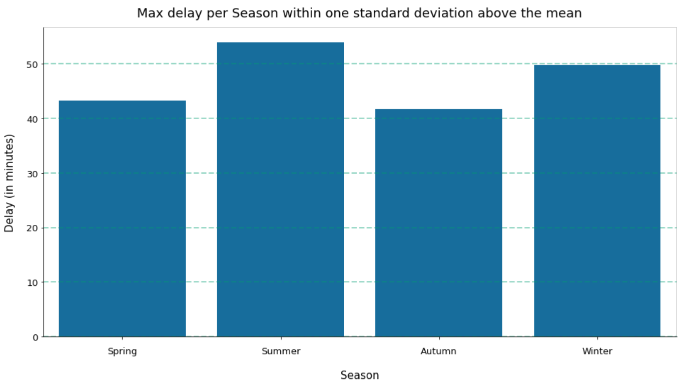


Figure 7:Max delay per season in Python

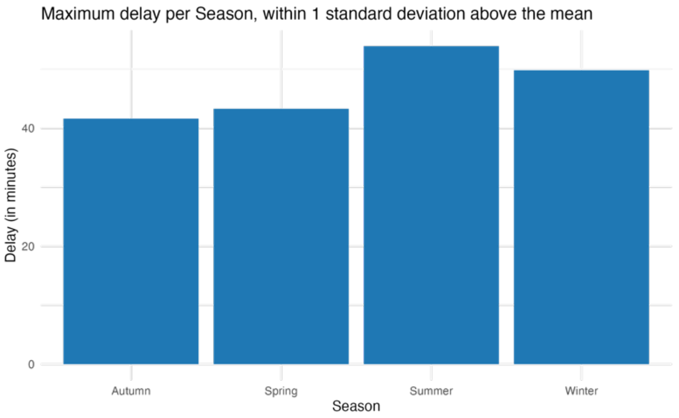


Figure 8:Max delay per season in R

Season	Average Delay	Maximum Delay (1 standard deviation above the mean)	Number of flights
Autumn	6.918309	41.612120	1460459
Spring	7.950428	43.273160	1455009

Since the maximum delay within 1 standard deviation above the mean for Autumn is the lowest maximum delay from all the seasons it can be said that Autumn is when passengers are most likely to minimze their arrival delay.

Question 2: Do older planes suffer more delays?

The 'Age' coloumn which was created in the preprocessing stage indicates the age of each aircraft. Thus the the whole dataset was grouped by each unique age value, and the mean arrival delay and standard deviation of each age value was calculated. The calculated standard deviation was also used to calculate the maximum delay within 1 standard deviation above the mean.

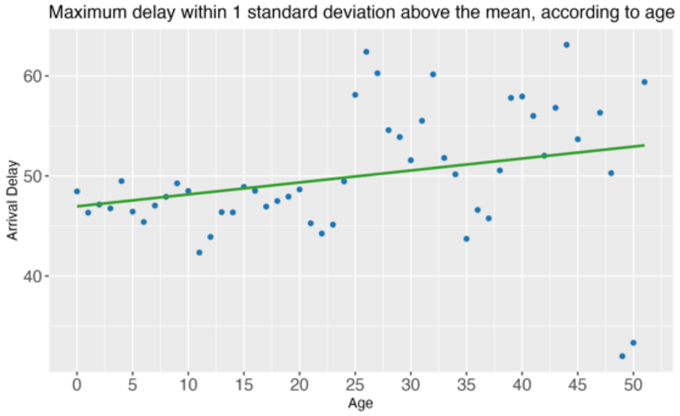


Figure 10:Max delay according to plane age

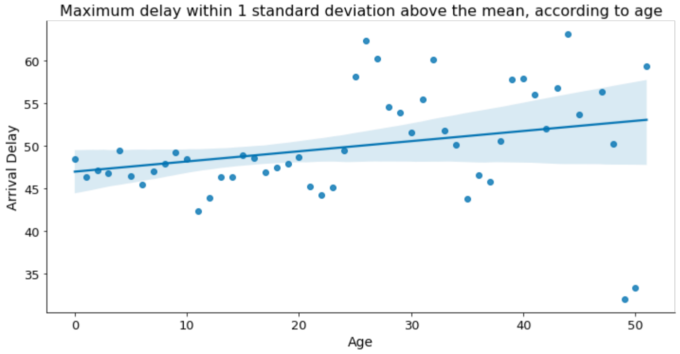


Figure 11:Max delay according to plane age

To further support the above evidence, it was decided to classify the aircrafts as follows.

Age of Aircraft	Classification
>= 25 years	Old Aircraft
< 25 years	New Aircraft

Thereafter, the mean delay and maximum delay within 1 standard deviation above the mean was calculated for both old aircrafts and new aircrafts.

	Avg. Delay	Max Delay
Old Aircraft	11.38454	54.79458
New Aircraft	9.476707	47.07867

The above figures and values further support the fact that the older the aircraft the more likely it is to experience larger arrival delays at its destination airport.

Question 3: How does the number of people flying between different locations change over time?

To explore the change in the number of people flying between different locations, it would be most ideal if there was data relating to the number of passengers. However, since we do not have access to such data, we will be carrying out this analysis by using the number of flights as a proxy variable for the number of passengers under the assumption that the number of flights and number of passengers are directly proportional and positively related.

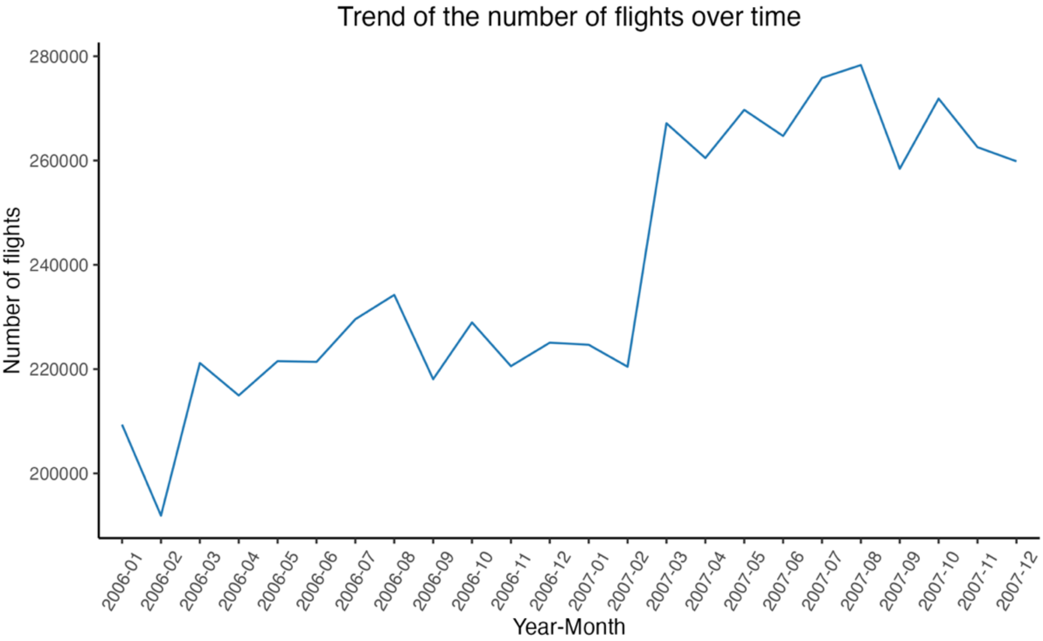


Figure 12:Trend of flights over time in Python

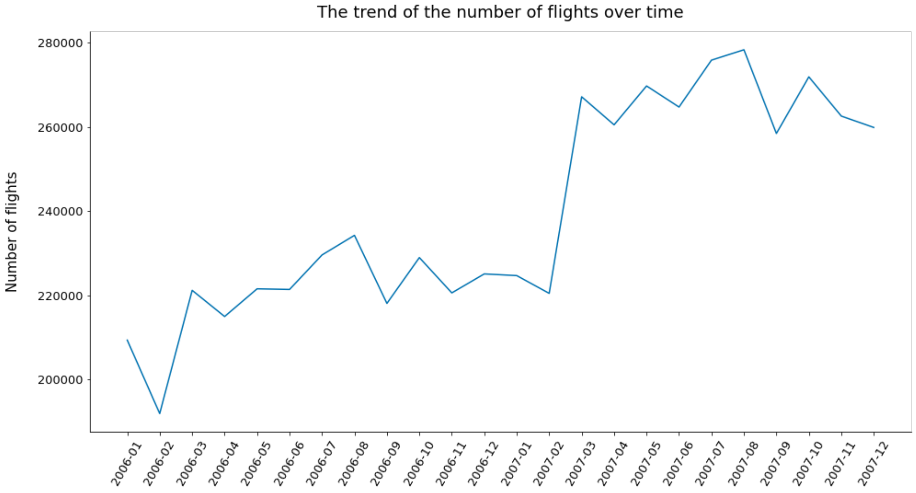


Figure 13:Trend of flights over time in R

The time series graphs shown above indicate the number flights over the many months throughout the two calendar years of 2006 and 2007. It is observed that there is fluctuating trend in the number of flights throughout a calendar year, as there are peaks during the summer and winter vacation months and the lowest air traffic is observed during the month of February. Most notable however is the sudden increase in the number of flights in the first quarter of the year 2007.

Question 4: Can you detect cascading delays as failures in one airport lead to failures in another ?

An aircraft experiencing an arrival delay at one airport could result in the same aircraft experiencing a delay at its next scheduled destination airport as well.

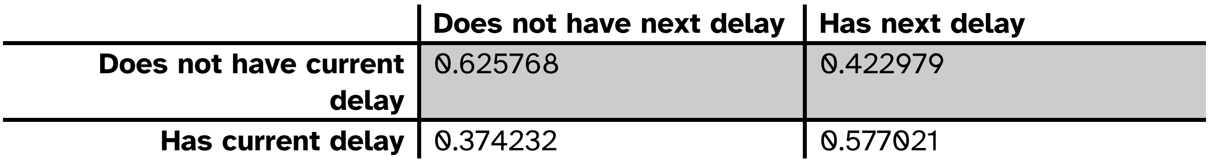
To identify whether such cascading delays were present, the data was grouped by the aircrafts' tail numbers and sorted by the date and time of the flights. Through this, it was possible to map the flights of each aircraft chronologically.

After performing the above mentioned sorting method, a new column called 'Next_delay', indicating the next arrival delay of each aircraft was created. Or in other words the arrival delay that an aircraft would experience at its next future destination airport was indicated by the new 'Next_delay' column.

This method causes a mismatch in the two columns as there is no Arrival delay value for the chronologically last flight at its hypothetical next destination, causing it to take on a null value and it was dropped.

The 'ArrDelay' (arrival delay) and 'Next_Delay' columns were encoded into 2 new columns to represent whether an aircraft had a current delay and/or has next delay.

A contingency table was created with the two newly encoded columns to check the frequency of each situation a plane could experience in terms of cascading delays.



	Does not have next delay	Has next delay
Does not have current delay	0.625768	0.422979
Has current delay	0.374232	0.577021

It can be seen that approximately 57.7% of flights that had a delay in the current airport will also experience a delay in the next destination airport and only 37.4% of the flights that had a delay in the current airport were able to reach their future destination on time.

This indicates that arrival delays experienced by aircrafts at their current destination can result in a future arrival delay for the same aircraft at its next destination.

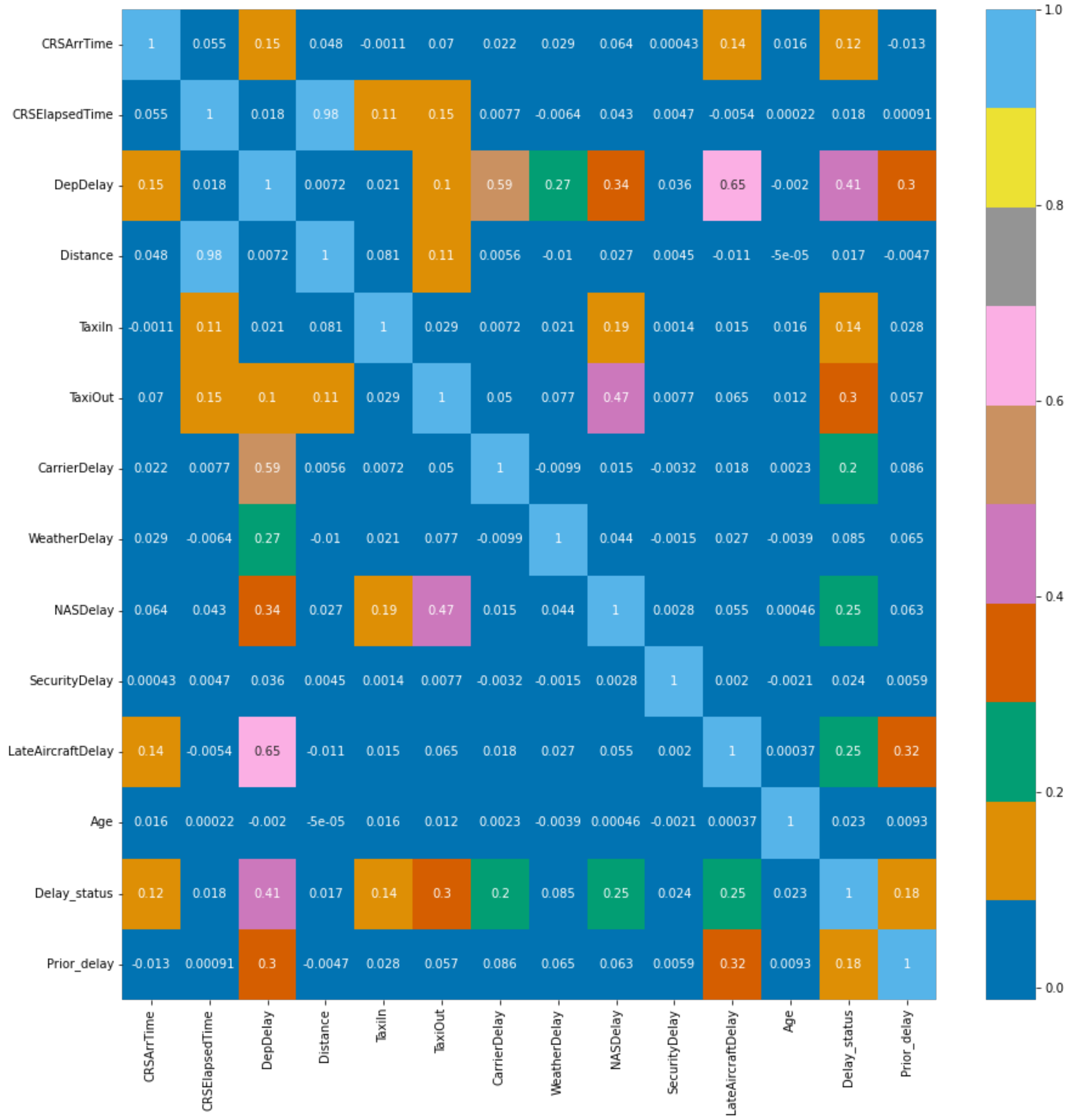
Question 5: Use the available variables to construct a model that predicts delays

As revealed in prior questions, arrival delays are possible under all circumstances. However, if we had the ability to predict the possibility of delays, we can increase the likelihood of experiencing smaller delays while increasing the likelihood of not experiencing any delay at all. This goal can be assisted through the help of machine learning. In this situation, it was decided to treat this problem as a classification problem. The reason being that predicting the possible presence of a delay or in other words the delay status; can be encoded into binary values with '1' being that a delay will occur and '0' being that a delay will not occur. To indicate this the 'Delay_status' column was created in the data preprocessing stage, where all delays greater than 0 minutes were classified as '1' and all other delay values were classified as '0'. As such it was decided to use a logistic regression machine learning model to predict the possibility of delays occurring.

In the previous question it was observed that arrival delays experienced by aircrafts at their current destination can result in a future arrival delay for the same aircraft at its next destination. As such it was assumed that each plane's arrival delay at its previous destination prior to its current flight, could be a potential contributor to predict the flight's arrival delay. A new column called 'Time_slot' was created which contained the hours of the scheduled departure time column, grouped into 3 time slots, of 8 hours each. This was to encode the scheduled departure time into a more manageable categorical variable of only 3 dummy variables for the regression model. The days of the week were initially represented through integers from 1 to 7 which represent Monday to Sunday, and so they were mapped to the actual days of the week in order to be encoded as dummy variables. The months of the year were mapped to the 4 seasons of the year to make it a more manageable categorical variable of only 4 dummy variables.

Columns such as 'Year', 'Month', 'DayofMonth', 'DayOfWeek', 'CRSDepTime', 'ArrDelay', 'Manufactured_year', and 'Year_month' were dropped as they are already captured or encoded in other columns. 'TailNum', 'UniqueCarrier', 'FlightNum', 'Origin', 'Dest' and 'Time_bin' columns were dropped as they had too many unique categorical variables to be encoded as dummy variables for the regression, as this will give rise to the curse of dimensionality and also result in the overfitting of the model. Meaning as more features are added to the model the model complexity increases, leading to requiring more data to get a more accurate model and at the same time the model will become to specified to the training data set and will not be general enough to perform well on other data sets. Columns such as 'DepTime', 'ActualElapsedTime', 'AirTime', 'ArrTime' were dropped as they contained time values that have already passed and thus already taken place and already captured in the arrival delay values, in some shape or form. As an example, in the case of 'Delay_status' it contains binary values that encode arrival delays greater than 0 and equal and lesser than 0, while arrival delay itself is calculated by subtracting the actual arrival time from the scheduled arrival time.

Dummy variables of the 'Time_slot', 'Day', 'Season' columns were generated and added as features of the target variable. A correlation matrix was plotted for all the numerical variables, to check the correlation between each variable. The departure delay column ('DepDelay') was selected as a feature as it had the highest correlation the target variable. The 'TaxiOut' column was selected as a feature as well, as it had the second highest correlation with the target variable, while having very low correlation with the departure delay column. No more columns were selected as most of them have very low correlation with the target variable or have too high of a correlation with the already selected features.



	CRSArrTime	CRSElapsedTime	DepDelay	Distance	TaxiIn	TaxiOut	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay	Age	Delay_status	Prior_delay
CRSArrTime	1	0.055	0.15	0.048	-0.0011	0.07	0.022	0.029	0.064	0.00043	0.14	0.016	0.12	-0.013
CRSElapsedTime	0.055	1	0.018	0.98	0.11	0.15	0.0077	-0.0064	0.043	0.0047	-0.0054	0.00022	0.018	0.00091
DepDelay	0.15	0.018	1	0.0072	0.021	0.1	0.59	0.27	0.34	0.036	0.65	-0.002	0.41	0.3
Distance	0.048	0.98	0.0072	1	0.081	0.11	0.0056	-0.01	0.027	0.0045	-0.011	-5e-05	0.017	-0.0047
TaxiIn	-0.0011	0.11	0.021	0.081	1	0.029	0.0072	0.021	0.19	0.0014	0.015	0.016	0.14	0.028
TaxiOut	0.07	0.15	0.1	0.11	0.029	1	0.05	0.077	0.47	0.0077	0.065	0.012	0.3	0.057
CarrierDelay	0.022	0.0077	0.59	0.0056	0.0072	0.05	1	-0.0099	0.015	-0.0032	0.018	0.0023	0.2	0.086
WeatherDelay	0.029	-0.0064	0.27	-0.01	0.021	0.077	-0.0099	1	0.044	-0.0015	0.027	-0.0039	0.085	0.065
NASDelay	0.064	0.043	0.34	0.027	0.19	0.47	0.015	0.044	1	0.0028	0.055	0.00046	0.25	0.063
SecurityDelay	0.00043	0.0047	0.036	0.0045	0.0014	0.0077	-0.0032	-0.0015	0.0028	1	0.002	-0.0021	0.024	0.0059
LateAircraftDelay	0.14	-0.0054	0.65	-0.011	0.015	0.065	0.018	0.027	0.055	0.002	1	0.00037	0.25	0.32
Age	0.016	0.00022	-0.002	-5e-05	0.016	0.012	0.0023	-0.0039	0.00046	-0.0021	0.00037	1	0.023	0.0093
Delay_status	0.12	0.018	0.41	0.017	0.14	0.3	0.2	0.085	0.25	0.024	0.25	0.023	1	0.18
Prior_delay	-0.013	0.00091	0.3	-0.0047	0.028	0.057	0.086	0.065	0.063	0.0059	0.32	0.0093	0.18	1

Figure 14:Correlation matrix

The data was split into 75% for training the logistic regression model, and 25% for testing the trained model. The dataset was then scaled, scaling is a technique that brings the magnitude of all the features onto the same standardized scale, thereby making sure that features with inherently larger numeric values do not hold greater influence over the model and result in a biased model.
As an example: if we were to build a model that predicts a person's gender using height in centimeters and weight in kilograms as features without scaling them the model may be biased towards the height feature as height usually takes on larger 3-digit values while weight typically takes on smaller 2-digit values.

Then the logistic regression model was trained and built with the training data and test data. The number of entries for the binary values in the 'Delay_status' column was checked to see if there was an imbalance in representation of the binary values of the column. Since there is an imbalance between the two 'accuracy' of the model is not a good metric to evaluate the model's performance. As such the model was evaluated using a confusion matrix and classification report as well as a ROC curve.

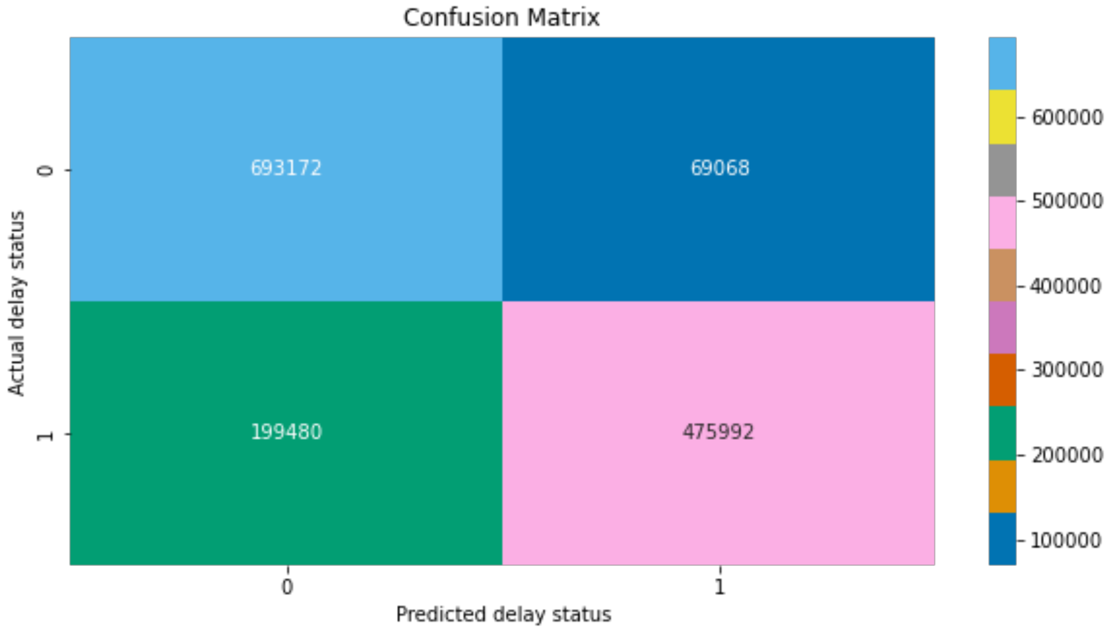


Figure 15:Confusion matrix

The confusion matrix visualizes the model's predicted outcomes against actual outcomes. A classification report was also obtained and it indicated that our model has a precision of 78% for predicting not delayed flights and 87% precision for predicting delayed flights. Precision in this case is the proportion of predictions that were actually correct.

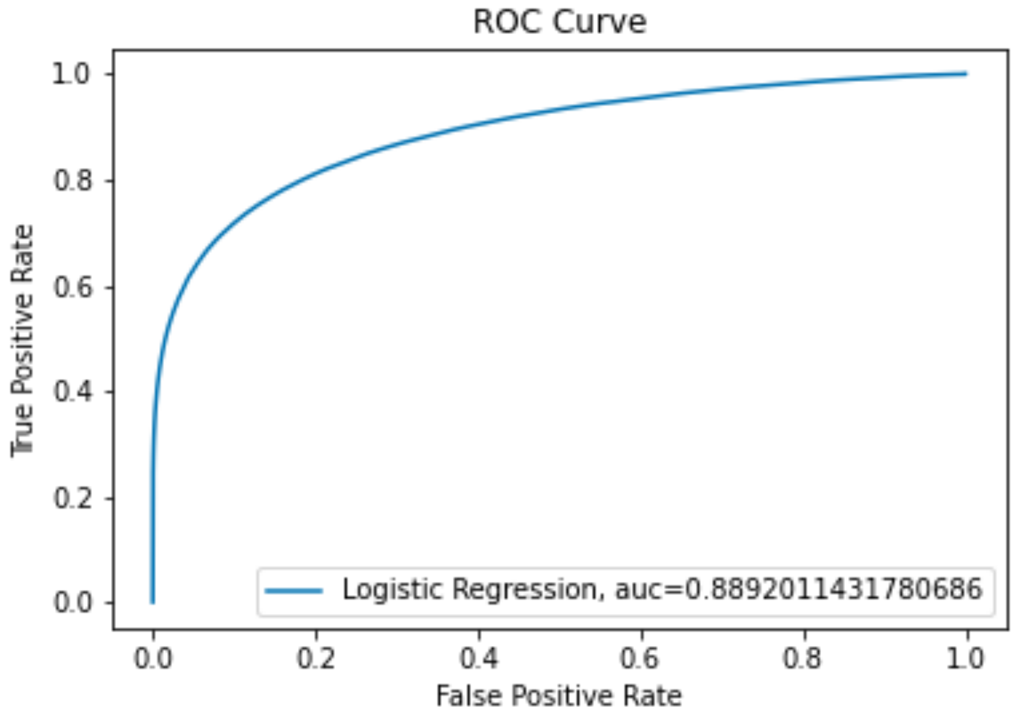


Figure 16:ROC curve

The ROC Curve visualizes the rate at which it correctly predicts both outcomes of the target variable, and in this case the model shows as that it has 0.889 area under the curve which is relatively high and that the model is much better than predicting by a 0.5 random chance.