PROGRAMMING FOR DATA SCIENCE

ST2195

Coursework

Name: Wee Jia Chyi Audrey

Student ID: 210342857

Table of Contents

1. Introduction

a. Aim

The Aim of this report is to produce findings through using R and Python to analyse flight arrival and departure details datasets extracted from Harvard Dataverse. From these discoveries airlines will be able to understand the trends of their aircrafts and audiences better.

b. Data

Throughout this report we will be using two years of flight data, 2005.csv , 2006.csv , airports.csv, plane-data.csv to answer the following five questions given.

c. Methods

To answer the questions, the Jupyter Notebook and R Markdown will be used to code in Python and R respectively. To visualise the data, various libraries such as seaborn and matplotlib will be used.

d. Preparation

Before the questions are analysed, the various datasets and libraries will need to be imported and installed. Following which, data cleaning is performed where the 'Cancelled' and 'Diverted' flights will be filtered out and unnecessary columns will be dropped.

2. Insights

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

This question will be broken down and answered in three sectors. The first being the best time of the day to fly, followed by best day of the week to fly and best time of the year to fly in order to minimise delays.

Since there are multiple variables which affects the arrival and departure timing, a 'TotalDelay' column is created where the 'ArrDelay', 'DepDelay', 'CarrierDelay', 'WeatherDelay', 'NASDelay', 'Security Delay', 'LateAircraftDelay' are summed together to take into account each factor that affects the delay timing of the flights.

To find the best time of the day to fly, we will be grouping the different times of the day within a 2 hour interval. This is formed by creating a new column, 'DepTime_2hours_interval' in the 'data_flights' data frame and binning the times in the 'DepTime' column through using the specific bin edges and bin labels with the help of the 'pd.cut' function to perform the binning operation. This results in the formation of a new column 'DepTime_2hours_interval' in 'data_flights' which contains the bin labels that corresponds to the bin in where the time in 'DepTime' falls under.

Grouping the 'data_flights' data frame by 'DepTime_2hours_interval' and calculating the mean for 'TotalDelay' for each group via the mean() method and filtering out values with a mean 'TotalDelay' less than zero since that will not be considered as a delay and will affect the results.

Following which a bar chart is plotted to visualise the mean 'TotalDelay' by 'DepTime_2hours_interval'.

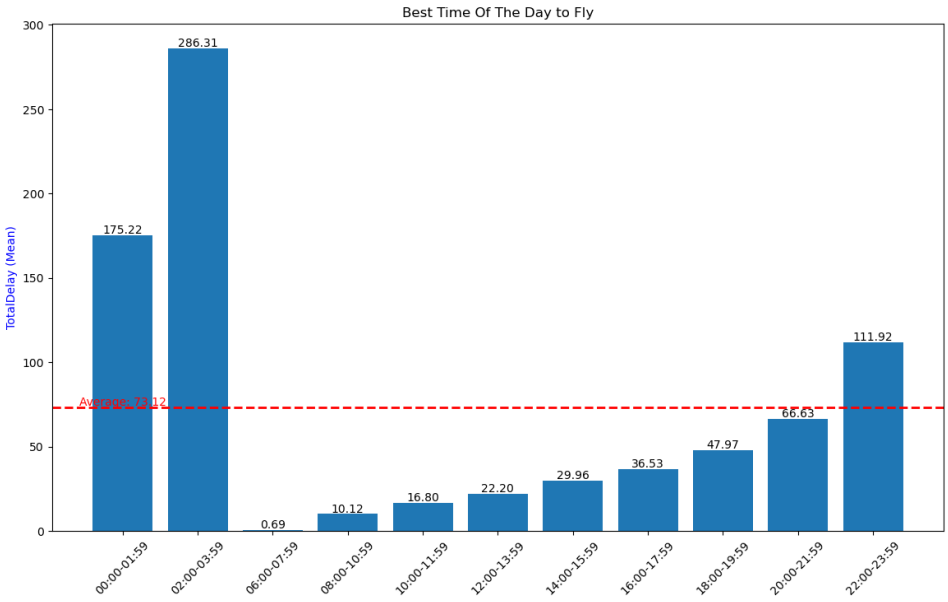

Figure 1: Bar Chart showing Mean Total Delay by Timings of the day

From Figure 1, it is discovered that between 06:00 – 07:59 is the best time to fly to minimise delays as the mean total delay for 06:00 – 07:59 is 0.69 minutes while the highest mean total delay is between 02:00 – 03:59 at an average of 286.31 minutes. In general, the mean total delay between 00:00 to 03:59 is the highest throughout the day while from 06:00 – 22:59, the mean total delay gradually increases. The average of the mean total delay is 73.12 minutes and from 06:00 to 21:59, the mean total delay falls below the average line, hence making that time frame a good alternative period to fly other than 06:00 – 07:59.

To find the best day of the week to fly, a new column, 'DayOfWeek_days' is created which contains each day of the week that corresponds to each numerical value, 1 to 7 in the 'DayOfWeek' column with 1 being Monday and 7 being Sunday.

Through grouping the data frame by ‘DayOfWeek_days’ and calculating the mean of ‘TotalDelay’ for each group and filtering out days with ‘TotalDelay’ at and below zero, a bar chat is plotted to visualise the mean ‘TotalDelay’ by ‘ DayOfWeek_days’ for the flights in ‘data_flights’.

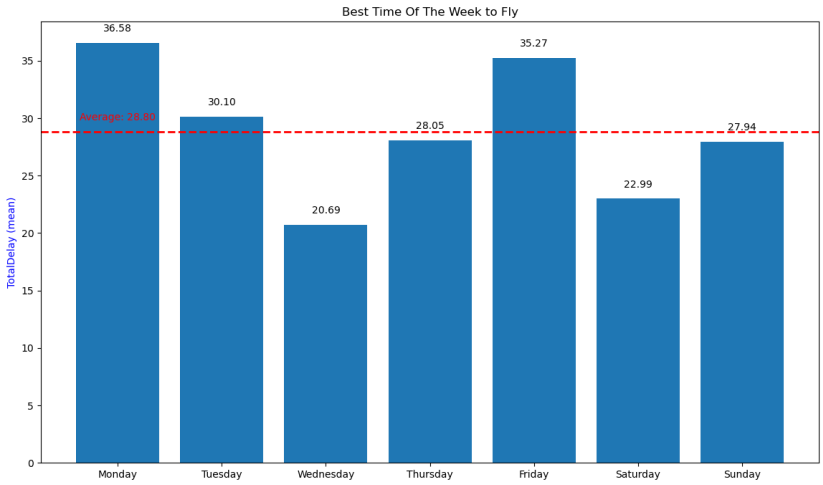

Figure 2: Bar Chart showing Mean Total Delay by Days of the Week

From Figure 2, Wednesday is the best day to fly at the lowest mean total delay amongst the week at 20.69 minutes while the highest mean total delay is 36.58 minutes on Monday. Overall, there are no extreme differences in terms of the mean total delay of the days in the week. The best day of the week to fly is ranked from the best to worst as follows, Wednesday, Saturday, Sunday, Thursday, Tuesday, Friday and Monday. The top 3 best days to fly in the week is Wednesday, Saturday and Sunday where their mean total delay falls below the average line of the mean total delay of 28.80 minutes.

For the best time of the year to fly, we will be using the months of the year to justify which month in the year is the best time to fly. Hence, a new column, ‘Month_Names’ is created where it contains the names of the months from January to December that corresponds to each value between 1 to 12 in the ‘Month’ column with 1 being January and 12 being December.

A bar chart is of mean total delay ‘TotalDelay’ by months ‘Month_Names’ is plotted through grouping ‘Month_Names’ and calculating the mean ‘ Total Delay’ of each group and filtering out the mean ‘TotalDelay’ which are less than zero to uncover which month in the year will be the optimal time to fly.

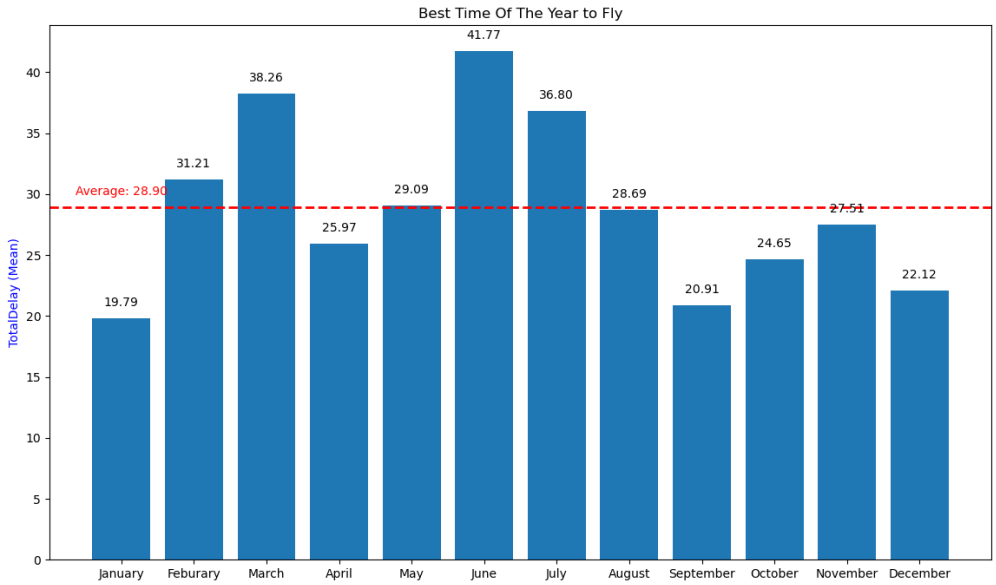

Figure 3: Bar Chart showing Mean Total Delay by Month

Figure 3 shows that January will be the best month of the year to fly with the lowest mean 'TotalDelay' at 19.79 minutes while the least optimal month to fly will be June where the mean 'TotalDelay' is at 41.77 minutes. Generally, the best periods to fly in the year will be from August to January as seen from Figure 3 as the mean Total Delays for these 6 months falls below the average total delay of 28.90 minutes.

In conclusion, the best time of the day, day of the week and time of the year to fly to minimise delays will be 06:00 – 07:59, Wednesday and January respectively as concluded from the bar charts in Figure 1 to Figure 3.

b. Do older planes suffer more delays?

As the data frame, 'data_flights' does not contain data on the age of the planes, the data frame needs to be merged with the 'planes' data frame base off the column 'tailnum' from 'planes' and 'TailNum' from 'data_flights'. This forms a new data frame, 'data_flights_planes' with the data needed to calculate the age of the planes. After renaming the 'year ' column to 'ManufacturedYear' and dropping the 'None' values in the column, the age of the plane is calculated in a new column, 'PlaneAge' , by subtracting the 'ManufacturedYear' from the 'Year' column where 'ManufacutedYear' represents the issue year of the plane and 'Year' represents the year of the flight.

To conclude if it is true that older planes suffer more delays, a line plot showing the mean total delay of flights for the different plane age groups is created.

This is done by firstly grouping the data by 'PlaneAge' and calculating the mean 'TotalDelay' for each group. After removing any groups with a mean total delay of zero or less, an array of integers with a length equal to the number of remaining age groups is created to be used as the x-axis in the following line plot. The mean 'TotalDelay' for each age group is then plotted against the corresponding 'PlaneAge'. The best fit trend line is also added to the plot by using a linear regression model.

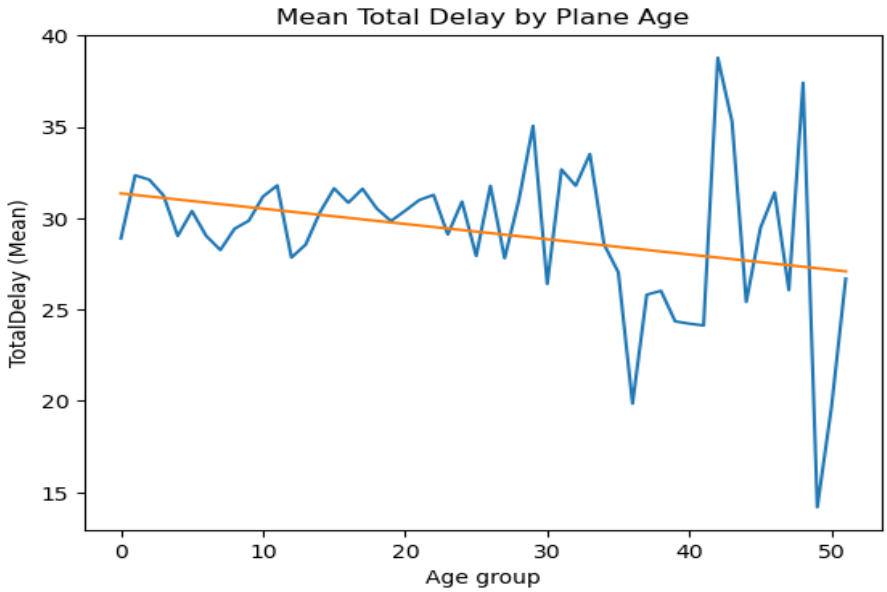

Figure 4: Line Plot Showing Mean Total Delay by Plane Age

Figure 4 shows that there is a inversely proportional relationship between mean 'TotalDelay' and 'PlaneAge' where the older the plane is, the lower the mean total delay. However, this can only be concluded based off the trend line as shown in Figure 4. From the Figure, it can be seen that the mean total delay faced by younger planes seem to be more consistent compared to the mean total delay faced by older planes where the line graph shows greater fluctuations as the age group of the planes increase.

In general, older planes do not necessarily suffer more delays. However, the greater the plane age, the more unpredictable it is to detect delays as the trends of total delay for planes gets increasingly inconsistent as the plane ages.

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

To gain more insights on the details of the origin and destinations of flights such as the city, airports and coordinates, the 'data_flights_planes' data frame is merged with the 'airports' data frame. This is done via merging the 'iata' code in 'airports' and 'Origin' in 'data_flights_planes' to form a new data frame, ' flights_planes_airports' with the addition of new columns 'Origin City', 'Origin State', 'Origin Lat', 'Origin Long'. Subsequently, 'flights_planes_airports' is merged again with 'airports' based off the 'iata' code in 'airports' and 'Dest' in 'flights_planes_airports' to create new columns on 'Dest City', 'Dest State', ' Dest Lat' and 'Dest Long' in the data frame.

Another two new columns, 'FlightRoutesCity' and 'FlightRoutesState' are created by concatenating the 'Origin City' and 'Dest City' as well as ' Origin State' and 'Dest State' columns together respectively.

As there are 52 states in the USA according to the data frame, plotting a Sankey diagram to see the flow of flights from one state to another would not be clear due to the saturation of data. Thus, the states will be classified and grouped into regions of the USA so that the diagram formed would be clearer to filter which the data that should be used to conclude how the number of people flying between locations changes with time.

The states will be classified into one of the six variables in the new 'Regions' dictionary; 'West', 'Midwest', 'Northeast', 'Southwest', 'Southeast', 'Others'. (Quai, N.D.). This is achieved by adding new columns, ' OriginRegion' and 'DestRegion' where each flights origin and destination state are used to determine it corresponding region using the 'Regions' dictionary created before. Additional columns are also create to concatenate the origin region and destination region.

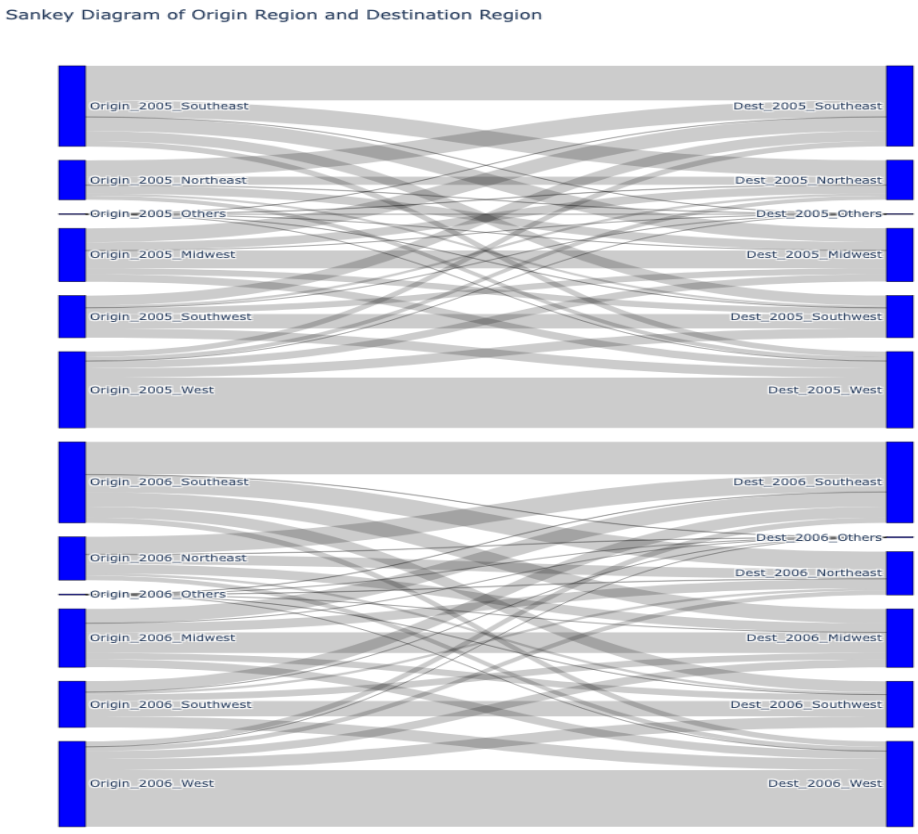

Figure 5: Sankey Diagram of Origin Region and Destination Region in 2005 and 2006

Next, two new data frames are created to separate the year 2005 and year 2006 data to form and extract the 'Year', 'OriginRegion' and 'DestRegion' columns into the data frames 'df' for 2005 and 'df1' for 2006. Both 'df' and 'df1' are

then grouped by the 'OriginRegion' and ' DestRegion' columns and a new column 'count' is formed by counting the occurrences of each combination. The resulting data frames are stored in the 'grouped' and 'grouped1' variables.

A Sankey diagram is formed which shows the flow of flights between the origin region and destination region of years 2005 and 2006.

Figure 5 shows that in both years 2005 and 2006, most flights still caters from one West Region to another West Region, followed by Southeast to Southeast in both years as well. This shows that over two years, most travellers within the USA still travels within the region. Thus, we will take the top 5 regions in the West to West regions to analyse flight patterns amongst states in the West Region

To do so, the 'flights_planes_airports' data frame is filtered by 'West – West' region from the 'FlightRouteRegion' columns and the top 5 states are computed by counting the number of flights which have occurred in both the origin and destination within the 'West' region (Figure 6).

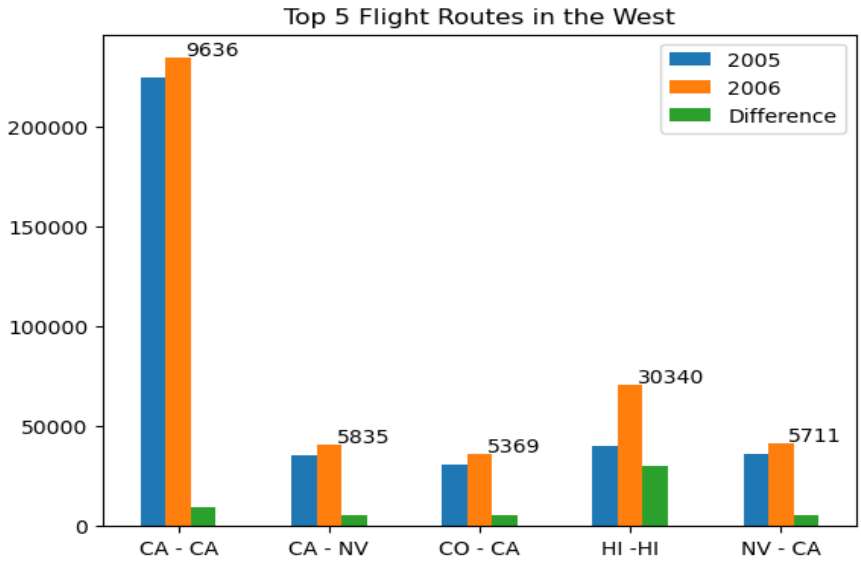

Figure 7: Bar Chart of Top 5 Routes in the West in 2005 and 2006

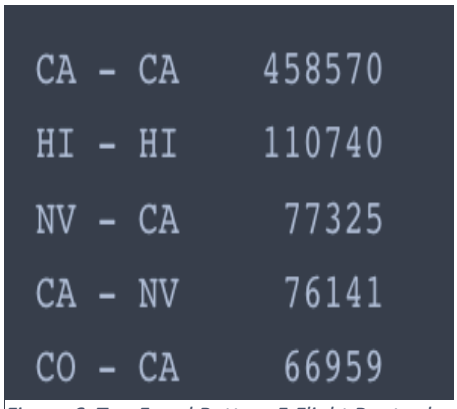

Figure 6: Top 5 and Bottom 5 Flight Routes by States in the West

After filtering the top 5 'FlightRouteStates' in the 'West – West' region from the combinations in Figure 6, a bar chart for the top 5 flights routes by state is created where the number of flights for each of the routes in the West region for 2005 and 2006 are compared.

Figure 7 shows the top 5 routes in the West region. With the labelled bar chart being the difference in the flight counts between 2006 and 2005. From Figure 7, it can be concluded that it is the most common for flights to be within the same state as the count for flights between CA – CA is the highest followed by HI – HI. Additionally, it can be seen that there is an increase in state to state flights from 2005 to 2006 from the differences in 2005 and 2006 where flights for HI – HI increased the most by 30340, followed by CA – CA where flights increased by 9636.

To see if the flight distance have changed overtime, the distance between each origin and destination airport for each flight in 'flights_planes_airports' data frame is calculated to create a new column, 'distance'. This is calculated

form the distance between two pairs of coordinates, which are extracted from the 'Origin Lat', 'Origin Long', ' Dest Lat', 'Dest Long' columns.

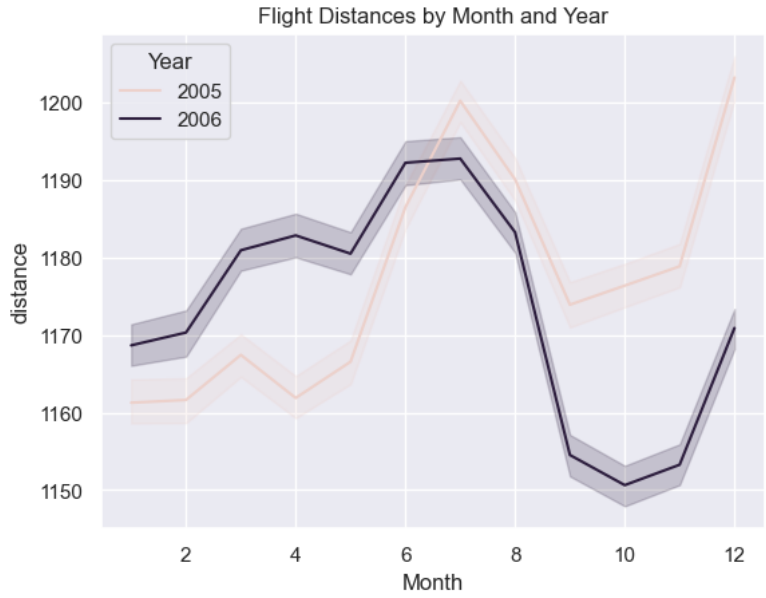

Figure 8: Flight Distances By Month and Year

Figure 8 shows a line plot where Month is plotted against distance and the two lines represents years 2005 and 2006. From the graph, it can be seen that across the year in 2005, the distance travelled generally increases while in 2006, the distance travelled generally decreases.

From Figure 8, it is observed that in both 2005 and 2006 have similar trends across the months in terms of the distance travelled. From month 1 of the year to the 7th month, the distances travelled within USA generally increases for both years. From month 7 to the 10th month, there is a downward trend in the distance travelled and in the last 2 months of the year, the distance travelled increases again.

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

To detect the cascading failures as delays in one airport that creates delays in the others, the 'airports' dataframe need to be merged with the 'flights_planes_airports' data frame based on the 'iata' column in 'airports' and 'Dest' column in 'flights_planes_airports'. The same is repeated to the Origin column in the 'flights_planes_airports' again after renaming the new columns to 'DestAirport' and 'DestCity' followed by 'OriginAirport' and 'OriginCity.

A new data frame 'cf' is created by selecting the 'Year', 'Month', 'DayofMonth', 'TailNum', 'ArrTime', 'DepTime', 'CRSDepTime', 'CRSArrTime', 'ArrDelay', 'DepDelay', 'OriginAirport' , 'DestAirport' from the 'flights_planes_airports' data frame. Next, a new column 'Date' is formed by selecting the columns 'Year', 'Month' and 'DayofMonth' from 'cf' and converts them to strings and concatenates the variables with '- ' and converts the resulting string into datetime format.

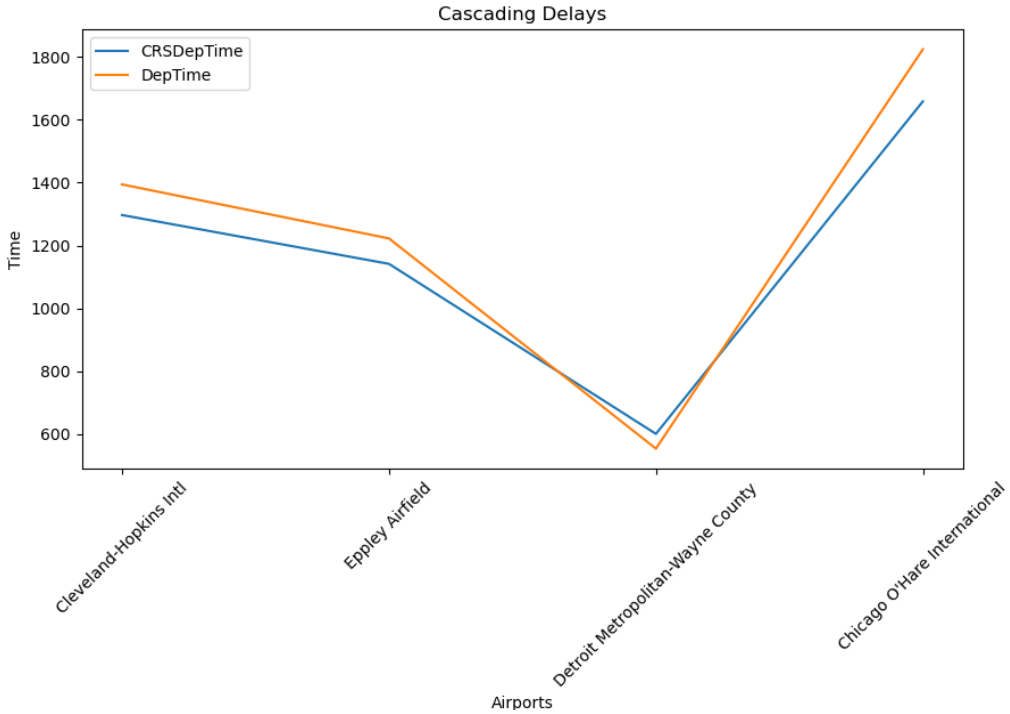

Figure 9: Line Chart showing Cascading Delays

A new data frame, 'cfN902UA' is created by filtering 'cf' to include only rows where the 'TailNum' column is equal to 'N902UA'. From looking at the counts of each date in 'cfN902UA', it can be seen that the date '2006-02-10' and '2006-12-31' has the highest count of flights in the day. Hence we will narrow down to analyse the flights in '2006-12-31' to detect cascading failures

By calculating the mean schedules depart time and departure time for each 'OriginAirport' in the data frame 'cfN902UA_2006_12_31', the mean time for each value is plotted against the airports.

Figure 9 shows that the departure time, 'DepTime' is above the schedules departure time 'CRSDepTime' for all of the airports except for Detroit Metropolitan-Wayne County. This shows that the departure time is delayed and not going according to the schedule and would affect subsequent scheduled flights in the other airports, hence creating delays.

Figure 10: Network Graph

Figure 10 shows a network graph where each row correpsonds to an edge of the graph with attirbutes 'OriginAirport', 'DestAirport' and 'ArrDelay. '

After filtering out ‘ArrDelay’ values equal or less than zero, the network graph show that from Chicogo O’Hare International and Cleveland-Hopkins and vice versa, there is an ‘ArrDelay’ or 72.50 minutes and from Eppley Airfield to Chicogo O’Hare International and vice versa, there is an ‘ArrDelay’ of 123.00 minutes. With an overall delay in these three airports, it is safe to conlcude that th 21.00 mintues delay in the one way flight from Chicogo O’Hare International to Antonia International is due to delays from the previous airports.

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

To construct a model which predicts delays, we will taking a sample of 1 million data from the ‘flights_planes_airports’ data frame and renaming the new sample as flights_planes_airports’_sample’. As there is no missing values, data cleaning is not required and a sample of 1 million can be taken.

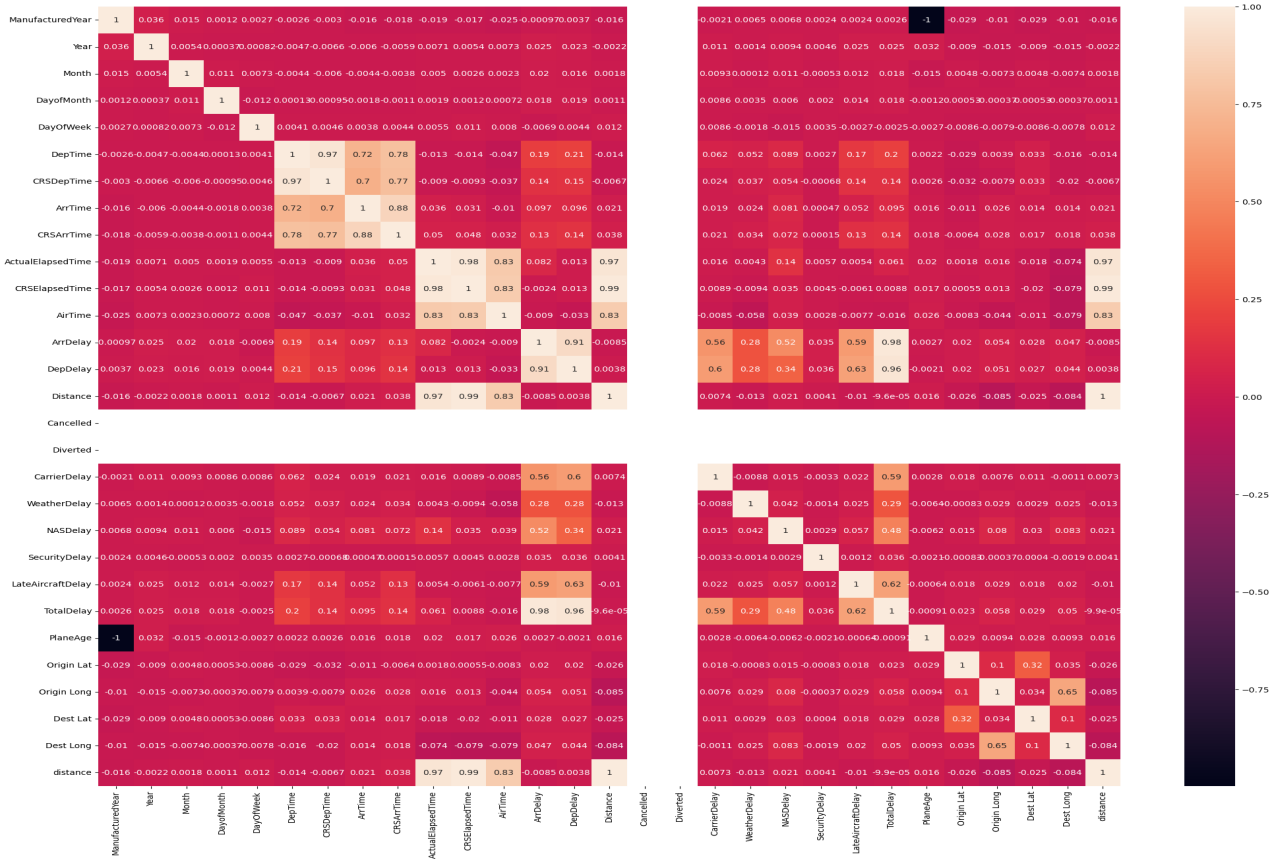

Figure 11: Correlation Matrix of 'flights_planes_airports'

By plotting a correlation matrix (Figure 11) on the variables in the ‘flights_planes_airports’ data frame, it is clearer to sieve out the variables which have high correlation to delays.

From the correlation matrix, 14 numerical variables,
'Year', 'Month', 'DayofMonth', 'CRSDepTime', 'CRSArrTime','DepTime','ActualElapsedTime', 'CRSElapsedTime', 'AirTime', 'ArrTime', 'DepDelay', 'ArrDelay', 'distance' and 'TotalDelay' are selected to build the models. With these numerical values, a Bayesian Linear Regression, Linear Regression, Ridge Regression, Lasso Regression and Random Forest models will be formed.

The ‘train_X’ data frame is created that includes the variables in ‘Numerical_Features’ and the ‘train_Y’ data frame is created where it only includes the ‘DepDelay’ variable from the ‘flights_planes_airports_sample’ which is the target of the regression model.

Following which, the data is randomly split into training and testing sets for the regression model. By using the ‘train_test_split’ function. The ‘train_X’ and ‘train_Y’ dataframes are then further split into four parts, ‘train_x’, ‘test_x’, ‘train_y’, ‘test_y’ which are the features for training set, features for testing set, target variable for training set and target variable for testing set respectively. The proportion of the dataset to include in the testing set is 30%

for ‘test_size’ and ‘random_state’ sets the seed for the random number generator ensuring that the same random split is achieved repeatedly.

Following which, each of the regression models and random forest model are fitted into the training data set to make predictions on the training and testing sets. The mean squared error(MSE) for the training and testing predictions and R-squared value is calculated.

	models	MSE Train Data	MSE Test Data	R Squared
0	Bayesian Linear Regression	1.749117e-18	1.227479e-18	1.000000
1	Linear Regression	5.562280e-27	5.552988e-27	1.000000
2	Ridge Regression	3.982409e-11	2.794684e-11	1.000000
3	Lasso Regression	6.126733e-03	6.236793e-03	0.999994
4	Random Forest	4.733698e-02	8.135252e-01	0.999740

Figure 12: Table of Models

From Figure 12, it can be seen that for the Bayesian Linear Regression, Linear Regression and Ridge Regression has a R Squared value of 1 which indicates that the model is a perfect fit for the data. The Lasso Regression and Random Forest model have a R-Squared value of 0.999994 and 0.999740 respectively which indicates a very good fit.

Noting that the models all have very low MSE values on both the training and test data, all of the models are effective in predicting the target variable ‘DepDelay’ in this data sample.

Based off the values in the table, the Bayesian Linear Regression, Linear Regression and Ridge Regression models are good choices for predicting the target variable since their MSE values are low and R Squared values are a perfect fit for the data.

To determine which model will be the best to predict delays for this data set, the advantages and disadvantages of Bayesian Linear Regression, Linear Regression and Ridge Regression will be taken into consideration.

Comparing the Bayesian Linear Regression model to the standard Linear Regression model, it is calculated using probability distributions instead of point estimates. Hence, resulting in the deduction that y is taken from a probability distribution. The advantages of the Bayesian Linear Regression is that it is easily understandable and provides a flexible and powerful frame work for modelling linear relationships by incorporating prior knowledge and giving way for uncertainty quantification. However, the Bayesian Linear Regression is not optima for large datasets due to its complexity which could lead it to be time-consuming and impractical. Hence since the data set analysed in this report is large, the Bayesian Linear Regression model is not optimal to help predict delays in flights.

Both Linear Regression and Ridge Regression are used to predict continuous dependent variables based on one or more independent variables.

Linear Regression hold the assumption that the terms are normally distributed and have a constant variance while Ridge Regression is made to shrink the coefficients size to avoid over fitting. Since the correlation matrix show high correlations between selected independent variables, the Ridge Regression Model is suitable to predict delays.

3. Bibliography

- Quai. (n.d.). *Welcome to quia*. Quia. Retrieved April 2, 2023, from https://www.quia.com/web
- ST2195 Study Guide
- *6 types of regression models in Machine Learning You should know about*. upGrad blog. (2022, December 1). Retrieved April 3, 2023, from https://www.upgrad.com/blog/types-of-regression-models-in-machine-learning/
- Dash, S. K. (2022, May 23). *Bayesian approach to regression analysis with python*. Analytics Vidhya. Retrieved April 3, 2023, from https://www.analyticsvidhya.com/blog/2022/04/bayesian-approach-to-regression-analysis-with-python/