Awarding Body

UNIVERSITY OF LONDON

Academic Direction

LSE THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

ST2195-PROGRAMMING FOR DATA SCIENCE

Analysis Report

Student no - 210516366

Degree Program – BSc Economics and Management

Table of Contents

1. Part I- Markov Chain Monte Carlo algorithm

In this part a Metropolis-Hasting algorithm was simulated to generate random numbers for the given distribution using R and Python. In the beginning the simulation was run for a standard deviation of 1 and 10000 Monte Carlo steps. Here, loops were executed from 1 to 10000 to generate 10000 samples including the 1st one. The mean of the sample was 0.10746 and the standard deviation of the sample was 1.37297 as in figure 01.

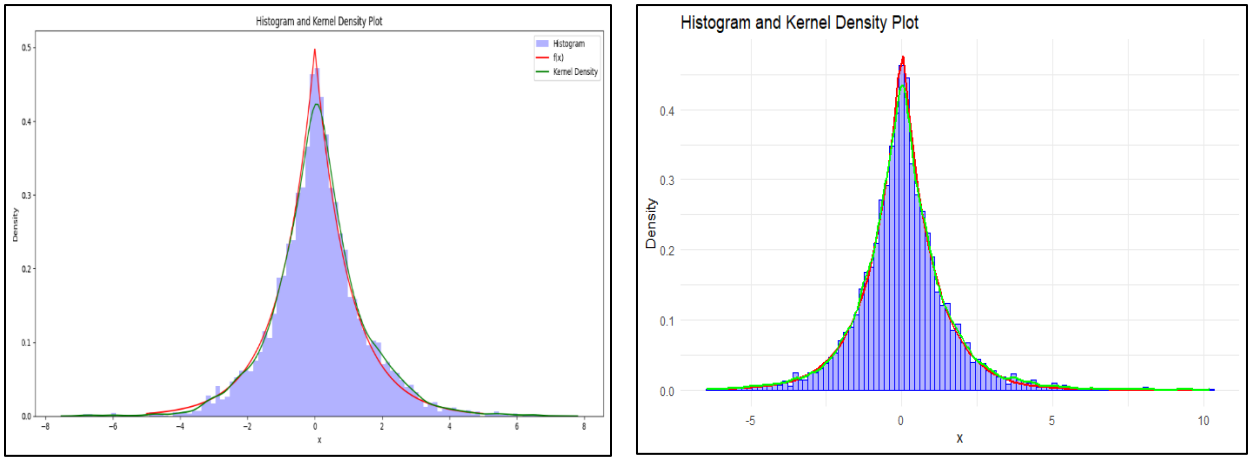

Figure 01: Output Part-I-(a) (Left: Python, right: R)

The program was then updated to generate more than one random number sequence. The newly given functions were defined and they were used to calculate the R̂ value, which gives us an idea about the convergence of the solution. The simulation was run for standard deviation of 0.001 and 2000 Monte Carlo steps. R̂ was calculated as 2.772. Which means the solutions of the simulator does not converge. But when the standard deviation was increased to 1, the R̂ value came to 1, which means the algorithm converges as in following figure.

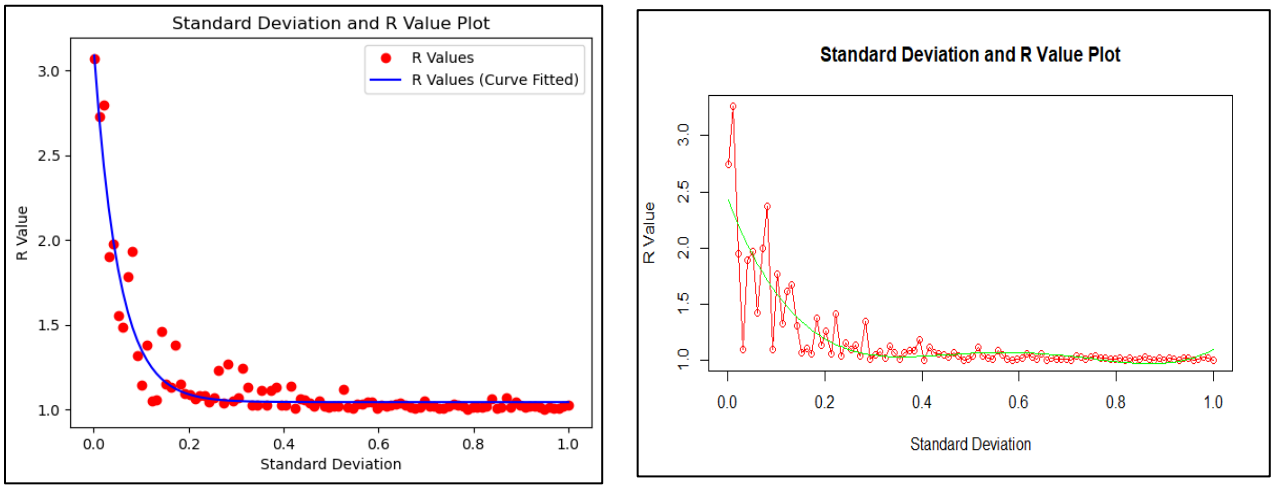

Figure 02: Output Part-I-(b) (Left: Python, Right: R)

2. Part II – Data Analysis

This section of the report delves into the data analysis and interpretation of a dataset of planes detailing planes' flights across various locations over time. It focusses on Departure delays from the origin and Arrival delays at the destination airports within the United States of America (USA).

The year '2006' and '2007' were selected for the analysis due to their substantial volume of data. Additionally, supplementary files containing airplane data were also utilized in the analysis process.

This segment of the report encompasses several key areas:

1. Data Cleaning Process: Detailing the steps taken to clean and prepare the dataset for analysis.
2. Optimal Time and Day for Minimizing Delays: Investigating the best times of day and days of the week to fly in order to reduce delays.
3. Impact of Aircraft Age on Delays: Exploring whether older planes experience more delays.
4. Model Development for Divert Predictions: Constructing a predictive model to anticipate flight diversions.

2.1. Data Collection and Clearing Process.

The two datasets, 2006 and 2007, that were provided to us were combined together to form a single dataset as they shared the same column structure. Unnecessary fields such as 'TaxiIn', 'TaxiOut', 'CancellationCode' were removed to streamline the dataset. Subsequently the 'TailNum' and the 'year' from the 'plane-data.csv' were right merged. Furthermore the 'year' field was renamed to 'ManuYear' in order to avoid confusion with 'Year' in the main dataset.

The coordinates of the airports were extracted from 'airports.csv' and merged into the dataset accordingly. All rows with null values, 'None's, 'NA's and negative delays were then removed. For part (a) and part (b) cancelled and diverted sets were removed and the data frame was saved as 'DelayData.csv'. For part (b) only cancelled sets were removed and was saved as 'DivertData.csv'.

Despite the identification of outliers, it was decided not to eliminate them, as doing so might result in the loss of crucial delay-related information. The preprocessing steps were documented and saved as 'Preprocessor' scripts in both .ipynb and .rmd formats. The data manipulation tasks were performed using the 'Pandas' and 'dplyr' libraries in Python and R, respectively.

2.2. The best time of day and day of the week to fly to minimize delays.

In this part we were asked to calculate the best times and days that can be used to minimize delays. Thereby, the arrival and departure delays were summed together to get an idea of the overall delay. For each best time of day and best day of week, the data frame was grouped by departure time and day. Then the average delay of each group was calculated. Results were interpreted using bar charts. The best date and time were identified by the minimum average delays and bar charts. The bar chart below represents the average delays based on the days of the weeks whereas the line graphs show the average delays based on the time of the day. The outputs were taken in both python and R as shown below.

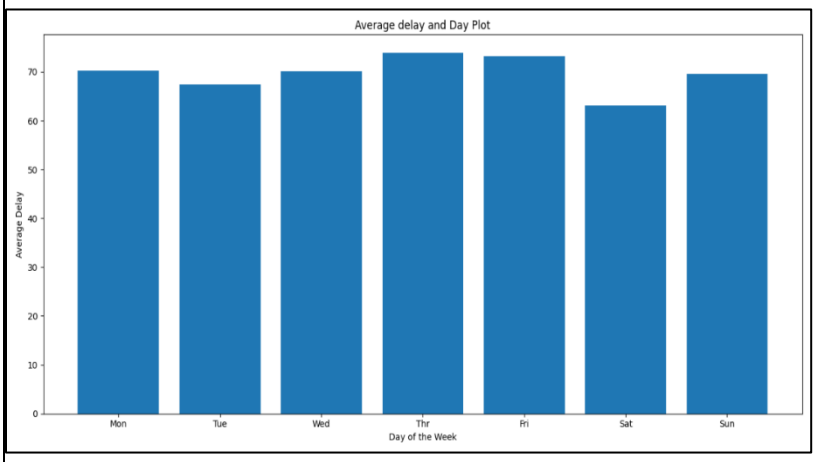

Figure 03: Average delay Vs Day of the week (Python)

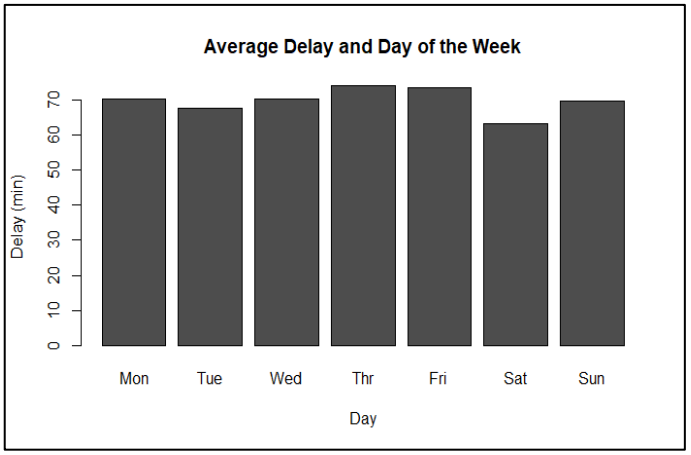

Figure 04: Average delay Vs Day of the week (R)

The above figures 03 and 04 that were obtained through both Python and R shows bar graphs that depicts the relationship between the days of the week and the average delays. It can be seen from the graphs that the day with the highest delays is Thursday whereas the day with the lowest average delays is Saturday. Furthermore, the program was also made to identify the day with the least amount of delays and identify it as the best day to minimize delays.

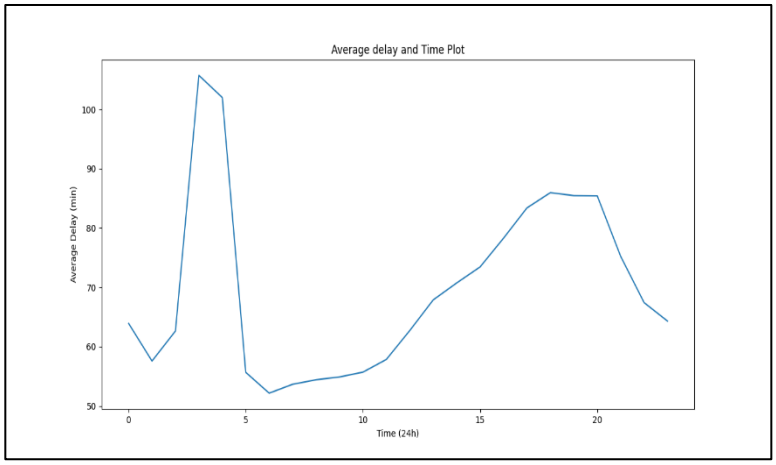

Figure 05: Average delay Vs. Time of the day (Python)

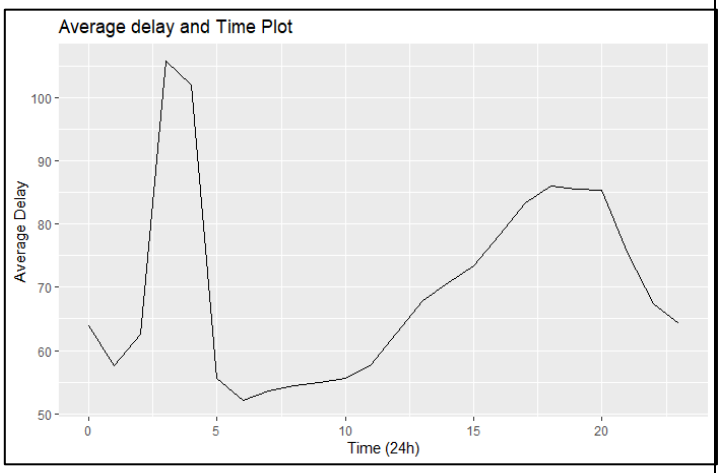

Figure 06: Average delay Vs. Time of the day (R)

As we can see in the above Line graphs, the time with the lowest delay is around 06:00am. Furthermore, the program was designed to output the best time of the day based on the lowest no. of delays which it gave as 06:00am and the time with the highest no. of delays which it gave as 03:00am.

Therefore, with the data that was obtained above we can conclude that the best day for a flight with the lowest delays was Saturday and the best time for a flight with the lowest delays was around 06:00am.

The dataset was used for the analysis of 4,493,512 records.

2.3. Whether older planes suffer more delays

In this part we were asked to calculate the delays based on the age of the airplanes.

The core analysis focused on examining the relationship between aircraft age and delays.

Average delay values were calculated for different aircraft ages, allowing for the observation of delay trends over the lifespan of an aircraft.

In order to get the age of the airplane, the manufactured year of each plane was subtracted from the year of the selected dataset. Then again, the data frame was grouped by age and the average delay was calculated as described in section 2.2. Furthermore, a quadratic curve fitting approach was employed to model the relationship between age and average delay, providing insights into the overall delay behavior as aircrafts age.

The results were obtained in both Python and R as follows:

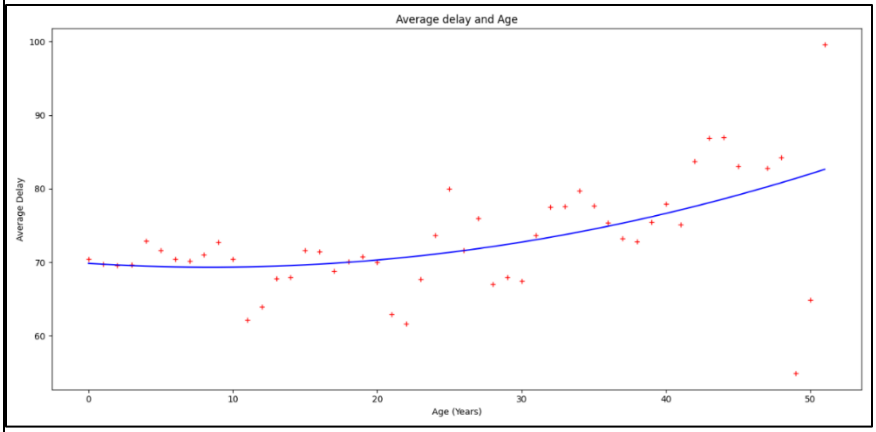

Figure 07: Average delay Vs. Age of the air plane (Python)

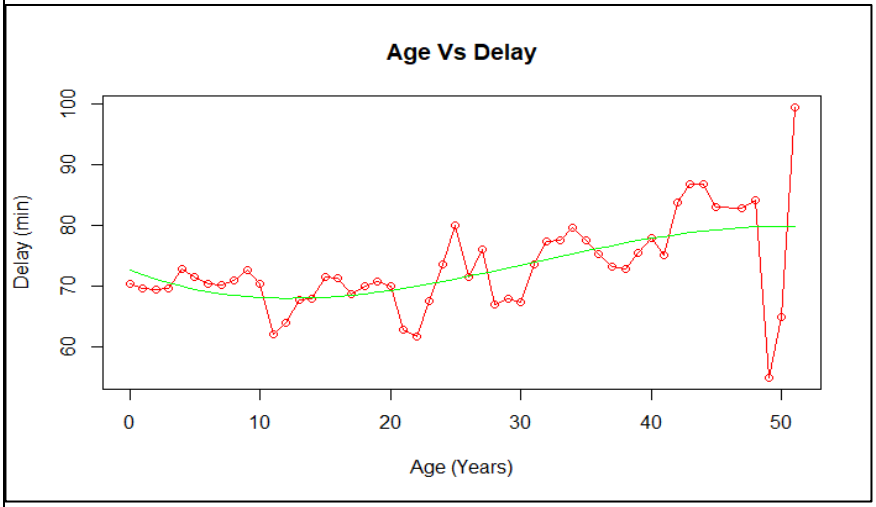

Figure 08: Average delay Vs. Age of the air plane (R)

As we can observe from the Figures 07 and 08, the average delays have increased as the planes age. However, an interesting observation was made regarding the periodic reductions in delay every 10 years of aircraft age. Despite these periodic reductions, delays continued to rise as the age of the aircraft increased. The periodic reductions in delay every 10 years could be attributed to maintenance schedules, repairs, or technological advancements in aviation. While these interventions might have temporarily alleviated delay issues, the underlying trend of increasing delays with age has persisted, possibly due to the cumulative effects of wear and tear on older aircrafts.
Therefore, we can conclude that older planes do indeed suffer more delays.

2.4. Constructing a model that predicts diverts

In this part of the coursework, we build a model to predict the diverts of flights. In order to consider which values were to be taken when building the model, a correlation plot was used to choose the variables with a higher correlation and were most related to diversions. The fields which were used to train the model included, 'DayOfWeek', 'CRSDepTime', 'CRSArrTime', 'UniqueCarrier', 'Distance' as well as origin and destination coordinates. The correlation coefficients were plotted for each year to visualize the relationships between these variables and flight diversions. This approach allows us to identify significant predictors and inform the construction of the predictive model.

The analysis involved preprocessing of the flight dataset to prepare it for modeling. Due to memory constraints (low memory issues), the dataset was down sampled from 14,308,548 records to 200,000 records (100,000 for each year). This down sampling allowed us for efficient utilization of computational resources while still maintaining a representative sample of the data.

The dataset was filtered and feature-engineered to include relevant variables such as carrier, departure time, arrival time, distance, and geographic coordinates. Categorical variables, such as 'UniqueCarrier', were one-hot encoded to facilitate modeling. Logistic regression models were trained and evaluated for the years 2006 and 2007.

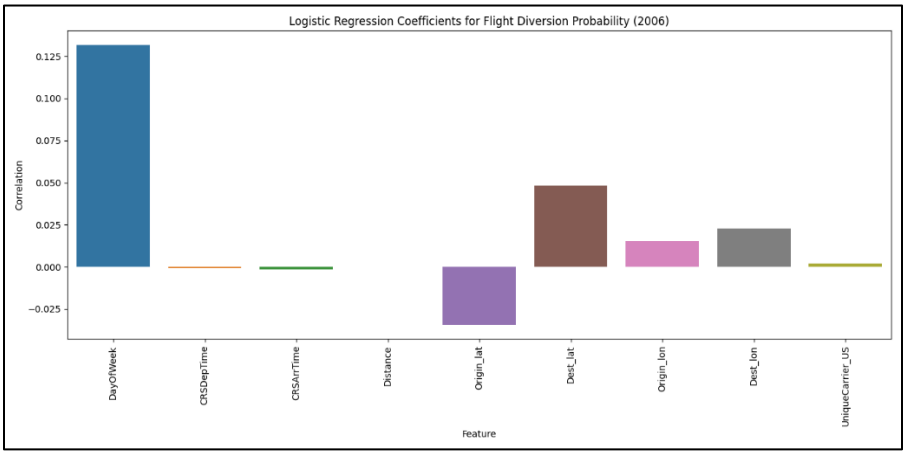

Figure 09: Coefficient of year 2006 (Python)

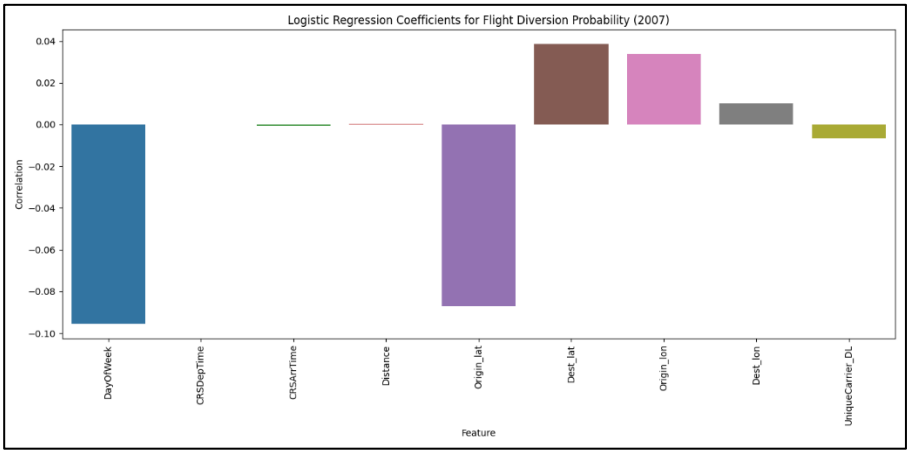

Figure 10: Coefficients of year 2007 (Python)

Comparative Analysis

Upon comparing the outputs of R and Python models, some notable differences were observed, particularly in the coefficients that were associated with the 'DayOfWeek' and UniqueCarrier'. These differences may have been as a result of the variations in the internal algorithms and implementations, especially in the treatment of one-hot encoded variables. Despite these discrepancies, both R and Python models identified 'Dest_lat' and 'Dest_lon' (destination coordinates) as significant predictors of flight diversion probability in both years.

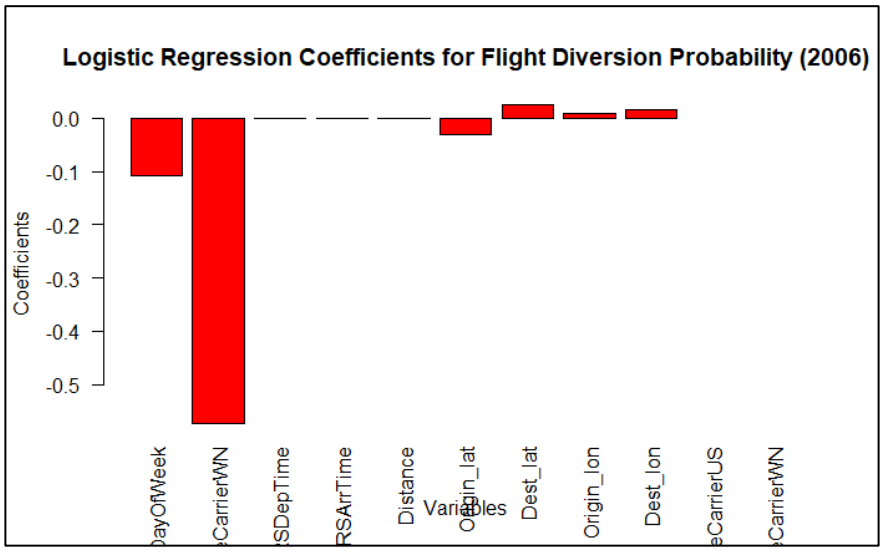

Figure 11: Coefficients of the year 2006 (R)

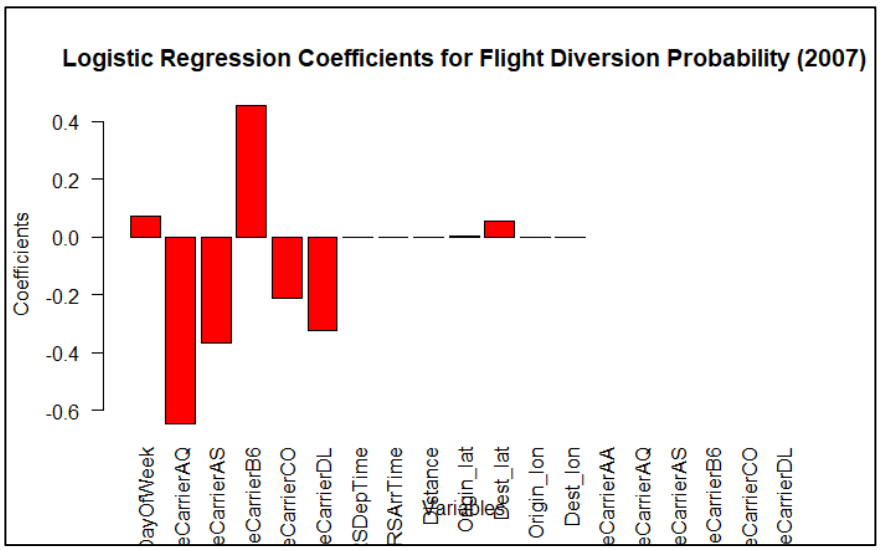

Figure 12: Coefficients of the year 2007 (R)

Key Findings

DayOfWeek Variation: The coefficient for 'DayOfWeek' exhibited contrasting trends between the two years, with a positive coefficient in 2006 and a negative coefficient in 2007. This suggests a shift in the influence of the day of the week on flight diversion probability over time.

Destination Significance: Both R and Python models consistently identified 'Dest_lat' and 'Dest_lon' as influential predictors of flight diversion probability across both years. This underscores the importance of destination coordinates in predicting diversion events.

In conclusion, this comparative analysis provides us valuable insights into the factors influencing flight diversion probability and the performance of logistic regression models in R and Python. Despite differences in outputs, common patterns and significant predictors were identified, offering robust insights for decision-making in aviation operations and risk management.

3. Reference

[1] Data expo 2009: Airline on Time Data (2008) *Harvard Dataverse*. Available at: https://doi.org/10.7910/DVN/HG7NV7.

[2] DPLYR (no date) *RDocumentation*. Available at: https://www.rdocumentation.org/packages/dplyr/versions/1.0.10

[3] GeeksforGeeks (2024) *Logistic regression in machine learning*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/understanding-logistic-regression/

[4] Pandas documentation# (no date) *pandas documentation - pandas 2.2.2 documentation*. Available at: https://pandas.pydata.org/docs/