



UNIVERSITY OF  
BIRMINGHAM

# Project Report

## Flight Data Visualization Web Application: Combining Aesthetics & Machine Learning

Xingyu Wang

Student ID: 2198771

BSc of Computer Science

Supervisor: Dr. Paul Levy

Inspector: Dr. Ian Batten

School of Computer Science

University of Birmingham

Word Count: 6071

Oct. 2022 - Apr. 2023

## Introduction

Air travel is an essential mode of transportation for millions of people around the world. Choosing the best airline is a crucial decision that has a huge impact on the entire travel experience. However, obtaining comprehensive and understandable information to make an informed choice can be a challenge. The Civil Aviation Administration (CAA) provides a large amount of flight information on its official website, including the place of departure and landing of the flight, the delay time and proportion of the flight, and the data set is compiled from the statistical reports of various airports. However, these valuable data are presented in the form of CSV or PDF files, which makes it troublesome for users to sift through and interpret the large amount of information.

Recognizing this challenge, the goal of this project was to create a flight data visualization web application that provides users with an intuitive and insightful way to access key information such as flight delays and airline complaints. Presenting data through visually appealing and easy-to-understand images, the web app aims to help users choose the airline that best suits their travel needs.

To achieve this, the project focused on addressing challenges in web development, design aesthetics, application of machine learning to predict airline delays, and incorporated user feedback to ensure continuous improvement of the webpage.

The main goal of this project is to:

1. Create a visually appealing and user-friendly data presentation web page.
2. Efficiently visualize various types of flight data.
3. Incorporate machine learning algorithms to predict airline delays.
4. Collect and implement user feedback to continuously improve web applications.
5. By transforming CAA's raw data into an interactive and engaging visual experience, the web application aims to empower users with the information necessary to make informed decisions when selecting an airline for their travel requirements.

A comprehensive web application is built utilizing cutting-edge frameworks for data processing and web visualization. This report delves into the project's journey, covering aspects such as research on data visualization techniques and methodologies, the gathering and enhancement of functional and non-functional requirements, the implementation of machine learning algorithms, software architecture design, and a thorough evaluation of the final application.

## Project platform

CPU: Intel(R) Core (TM) i7-9750H

GPU: 1650 Laptop

RAM: 16GB

Operating system: Windows 10 Home x64

## Literature Review

### **Integrating Machine Learning into Visual Analytics**

A. Endert et al. (2017) conducted a comprehensive survey of machine learning methods and visual analytics systems that effectively integrate machine learning. They identified a set of opportunities to further the integration between these two scientific areas, such as formalizing and establishing steerable machine learning, providing coupled interaction and visualization methods that offer substantially more advanced user feedback, and better determining how tasks should be divided between humans and machines, possibly in a dynamic manner. They also discussed the potential for developing considerably more powerful visual analytic systems with integrated machine learning using recent models and frameworks.

### **Factors Influencing Flight Delays**

Zámková, Prokop, and Stolín (2017) identified that factors contributing to delays included passengers, baggage handling, aircraft supply companies, maintenance and defects of the aircraft, operational management of human resources, air traffic control, and restrictions at airports. They noted that technical deficiencies and maintenance of aircrafts were more frequent causes of long delays, often occurring at night. Furthermore, previous delayed flights were found to cause long delays, with the frequency of these longer delays increasing significantly during the day (Zámková, Prokop, & Stolín, 2017).

Carvalho et al. (2021) identified six main perspectives for analysing flight delays: arrival, departure, propagation, airline, airport, and air system. Data sources were categorized based on airlines, airports, public agencies, weather databases, and open data initiatives. In terms of data pre-processing, Carvalho et al. (2021) observed the general application of data cleaning, integration, reduction, and transformation. They also noted the use of various data analytics methods, such as classification, clustering, machine learning, network analysis, pattern mining, regression, and statistical analysis.

Zámková et al. (2017) concluded that the most critical problem was chained delays triggered by the delays of previous flights, suggesting that lending a spare aircraft, especially during the peak season, could help alleviate this issue. They also recommended optimizing the supply of spare parts, improving logistics, and ensuring high-quality and timely maintenance of aircrafts. Additionally, they emphasized the importance of regular training for technicians, company traffic controllers, and adopting the most recent findings to improve the process.

### **Gradient Boosting for Function Approximation**

Friedman (2001) presented specific algorithms for various loss functions, such as least-squares, least-absolute-deviation, and Huber-M loss functions for regression, as well as multi-class logistic likelihood for classification. The author also derived special enhancements for cases where the individual additive components are regression trees and introduced tools for interpreting such "TreeBoost" models.

The gradient boosting of regression trees, as described by Friedman (2001), results in competitive, highly robust, and interpretable procedures for both regression and classification. This approach is particularly suitable for mining less-than-clean data, making it relevant to my project's goal of predicting flight delays using the LGBM algorithm.

### **Data Visualization Frameworks and Libraries**

Numerous data visualization frameworks and libraries are available, including open-source solutions and proprietary commercial products. Both types aim to create visualizations, but they cater to different use cases. Prominent examples include ECharts, a JavaScript-based data visualization library, and Bootstrap, a widely used responsive, mobile-first design framework. The choice between ECharts and Bootstrap depends on factors such as aesthetics, ease of use, and compatibility with other technologies.

ECharts, initially released in 2013, offers a variety of chart types and interactions, making it suitable for creating intricate visualizations. However, its aesthetics and interactivity may not

always meet user expectations. Bootstrap, on the other hand, focuses on providing a responsive and user-friendly layout that adapts to various screen sizes and devices. By combining the capabilities of ECharts and Bootstrap, developers can create visually appealing, interactive, and responsive data visualizations.

In my project, I initially used ECharts to create data visualizations for flight information. While ECharts provided a solid foundation, I found that its aesthetics and interaction effects were not sufficient for my needs. After reviewing successful projects that utilized both ECharts and Bootstrap, I decided to combine these two technologies for my data visualization website. This allowed me to create an attractive and user-friendly interface that effectively displayed flight delay and airline complaint data.

By using ECharts for the core visualization components and Bootstrap for the overall layout and responsiveness, I was able to achieve a balance between aesthetics and functionality. The combination of these two technologies proved to be an ideal solution for my project, resulting in a visually engaging and easy-to-use data visualization website that effectively presents flight information to users.

## Analysis & Requirements

### Data Clean Module

In my project, data processing played a crucial role in transforming the raw flight information obtained from CAA's official website into a visually appealing and user-friendly format. The initial phase involved cleaning and preparing the dataset, which was achieved using Python's Pandas library to handle missing values, inconsistencies, and inaccuracies. Once the data had been refined and structured, key metrics, such as flight delays, airline rankings, and complaint rates, were calculated and stored in a JSON file for easy access and retrieval.

### Data Visualisation System Functional Requirements

1. Accessible to all users, no registration required
2. All visualizations are dynamically generated using the processed flight data stored in the database.
3. Visualization dashboard can be re-sized and responsively adjusts layout for different screen sizes.
4. Data REST API endpoints consumed by visualizations should not include sensitive or confidential data.
5. Display of total flights for the month.
6. Pie chart visualization depicting the ranking of airlines in the number of flights.
7. Tool-tip appears on visualization, detailing the particular airline and associated number of flights when hovering over a section of the pie chart.
8. Vertical bar chart visualization for flight delay rate by duration of delay.
9. Horizontal bar chart visualization for top 10 flight cancellations.
10. Line chart visualization comparing the average delay time between predicted values and real values.
11. Table chart visualization displaying the top 10 complaints per million passengers.
12. Users can interact with the visualizations to gain deeper insights into specific data points or time frames.
13. Visualizations should be optimized for performance, ensuring quick load times and smooth user experience.
14. Users can filter visualizations by airline, time period, or other relevant factors to customize their view of the data.

15. The system should provide clear and concise documentation to help users understand how to navigate and interpret the visualizations effectively.

### Non-Functional Requirements

#### **Performance:**

- a. Visualizations should load quickly and efficiently.
- b. The system should be able to handle large volumes of data without any significant impact on performance.

#### **Scalability:**

- a. The system should be designed to handle an increasing number of users and data volume without performance degradation.
- b. The architecture should allow for easy addition of new features and visualizations in the future.

#### **Usability:**

- a. The user interface should be intuitive and easy to navigate.
- b. Visualizations should be clear and visually appealing, effectively communicating the underlying data.

#### **Responsiveness:**

- a. The layout of the dashboard should be responsive, adapting to different screen sizes and devices.
- b. Interactions with visualizations should be smooth and responsive, allowing users to explore data efficiently.

#### **Security:**

- a. The system should not expose any sensitive or confidential data through visualizations or API endpoints.
- b. Any user input or interactions should be properly validated and sanitized to prevent security vulnerabilities.

#### **Maintainability:**

- a. The codebase should be well-structured and modular, enabling easy updates and maintenance.
- b. Proper documentation should be maintained for both the code and system architecture to facilitate future development and troubleshooting.

#### **Reliability:**

- a. The system should be robust, ensuring that errors or failures do not impact the overall user experience.
- b. Any data processing or visualization updates should be completed accurately and consistently.

#### **Compatibility:**

- a. The system should be compatible with modern web browsers and operating systems.

b. Visualizations should render correctly and consistently across different browsers and devices.

**Accessibility:**

- a. The system should be designed with accessibility in mind, ensuring that users with disabilities can access and interact with the visualizations.
- b. Any text, colours, and interactive elements should adhere to accessibility standards and best practices.

**Testability:**

- a. The system should be designed to facilitate thorough testing of both functional and non-functional requirements.
- b. Test cases and procedures should be well-documented and maintained to ensure the ongoing quality and reliability of the system.

**Evaluability:**

- a. The system should be designed with appropriate criteria and methods to assess the success of the project.

## Web application Architecture

Software Architecture for Flight Data Visualization Project:

The software architecture for the Flight Data Visualization Project is designed to efficiently process, analyse, and present flight data through an intuitive and responsive web interface. The architecture is modular, scalable, and maintainable, facilitating future improvements and expansion. The main components of the architecture include:

Data Ingestion and Processing:

- a. ETL (Extract, Transform, Load) processes are employed to gather flight data from various sources, such as CSV files.
- b. Data pre-processing and cleaning are performed using Python and the Pandas library, ensuring consistent and accurate data.
- c. Processed data is stored in a JSON file format or a suitable database, such as PostgreSQL, for efficient retrieval and querying.

Frontend Application:

- a. A responsive and accessible web application is developed using HTML, CSS, and JavaScript, along with the Bootstrap framework for adaptive layout and design.
- b. Data visualizations are created using ECharts, a powerful and flexible charting library, to display flight data in various formats, such as pie charts, bar charts, line charts, and tables.

## Technologies & Frameworks

### Pandas

‘Pandas’ is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. (NumFOCUS)

Considering the need to process and analyse flight data efficiently, Pandas, a powerful Python library, was chosen for this project. The main strength of Pandas lies in its ability to handle large datasets and perform various data manipulation tasks with ease. It offers data structures like DataFrame and Series, which simplify the process of handling and cleaning data. In addition, Pandas provides a wide range of built-in functions for data aggregation, filtering, and transformation, which enabled us to prepare the data for visualization effectively.

By using Pandas, I was able to read and process the raw data from CSV and JSON files, perform necessary data cleaning tasks such as handling missing values and outliers, and aggregate the data to obtain meaningful insights. Furthermore, Pandas' seamless integration with other Python libraries, such as NumPy and SciPy, allowed us to perform advanced statistical analysis on the data when required.

### **JavaScript Object Notation (JSON)**

In the context of this project, JSON played a crucial role as a lightweight data interchange format. JSON is a text format that is completely language-independent but uses conventions familiar to programmers of the C family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. JSON is highly readable and easy for humans to understand, while also being efficient for machines to parse and generate.

One of the main benefits of using JSON in this project was its seamless integration with JavaScript, which was the primary language used for both front-end and back-end development. JSON facilitated the efficient storage and transmission of data between the server and the client, as well as between different components of the application. Its compatibility with various web APIs and libraries, such as ECharts and Bootstrap, made it an ideal choice for this project.

By utilizing JSON, I was able to store flight data in a structured and easily accessible format. This, in turn, allowed us to efficiently manipulate and process the data for visualization purposes.

### **Heatmap**

A heatmap is a data visualization technique that represents data in a matrix format, using varying colours or shades to indicate the values of individual cells. Heatmaps are particularly useful for displaying large datasets with multiple variables, allowing users to quickly identify trends, patterns, or correlations within the data.

In a heatmap, rows and columns represent different variables, while individual cells represent the intersection of these variables. The colour or shade of each cell corresponds to the value of the data point at that intersection. Typically, a colour scale is used to differentiate between high and low values, with one colour representing the lowest value and another colour representing the highest value. Intermediate values are depicted using a gradient of colours between the two extremes.

By using colours to represent data values, heatmaps provide a clear and intuitive way to display complex datasets, making it easier for me to identify trends and patterns at a glance.

### **ECharts**

ECharts served as a powerful, open-source charting and visualization library. Developed by Baidu, ECharts provides a comprehensive collection of customizable chart types, including bar, line, pie, scatter, map, and more. Its primary goal is to facilitate the easy creation of visually appealing and interactive charts, while maintaining high performance and flexibility.

One of the main benefits of using ECharts in this project was its seamless integration with JavaScript, which was the primary language used for front-end development. ECharts allowed us to create highly customizable and dynamic visualizations based on the flight data, making

it easier for users to understand and interact with the information. The library's extensive documentation and numerous examples enabled efficient development and quick adaptation to various project requirements.

By utilizing ECharts alongside other technologies such as Bootstrap for responsive design, I was able to create a visually appealing and user-friendly data visualization website. The combination of ECharts' rich features, ease of use, and customization options allowed us to present flight data in a clear and engaging manner, ultimately enhancing the overall user experience.

## **Bootstrap**

Bootstrap played a crucial role as a widely-used, open-source CSS framework designed to simplify and streamline the development of responsive, mobile-first web applications. Developed by Twitter, Bootstrap provides a comprehensive set of pre-built components, such as typography, forms, buttons, navigation, and various interface elements, along with a powerful grid system to aid in the creation of flexible and responsive layouts.

One of the key benefits of using Bootstrap in this project was its ability to significantly reduce the time and effort required for designing and developing the user interface. By leveraging the built-in classes, components, and utilities, I was able to create a clean, modern, and visually appealing design that seamlessly adapts to various screen sizes and devices.

In combination with ECharts for data visualization, Bootstrap contributed to the overall aesthetics and user experience of the website, ensuring that the visualizations and layout elements were consistent and responsive across different platforms. This allowed us to deliver a polished and user-friendly data visualization website, making it easy for users to navigate and interact with the displayed information regardless of their device or screen size.

## **Light Gradient Boosting Machine (LightGBM)**

LightGBM is a powerful open-source gradient boosting framework developed by Microsoft, designed to efficiently and scalably process large-scale datasets. As a tree-based learning algorithm, LightGBM builds decision trees by optimising loss functions and employing gradient boosting techniques to iteratively reduce prediction errors.

One of the main advantages of LightGBM over other gradient boosting frameworks such as XGBoost is its ability to handle large datasets more efficiently, thanks to its unique tree construction strategies including Gradient Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). These techniques help reduce computational complexity and memory consumption, enabling faster training and more accurate results even when dealing with large datasets. It is therefore ideally suited for working with datasets that have large amounts of flight data.

In the context of this project, LightGBM can be used to build predictive models based on the data collected. For example, it can be used to predict flight delays.

## **Project process & results**

### **Data processing and analysis Module**

To clean and prepare the data for visualization and analysis, I performed the following steps using the 'pandas' library in Python:

1. Read data: When using pandas to process data, I thought the data set was too large to be directly loaded into the cache at one time. use the 'pandas.read\_csv()' function with the 'chunksize' parameter. This parameter allows to read the data in smaller chunks,



processing one chunk at a time, thus avoiding the memory constraints. However, this method will result in slower reading data, so the best solution to avoid this problem is to use a device with larger memory.

2. Data inspection: I thoroughly inspected the dataset using methods such as 'info()' and 'head()' to understand its structure, identify any inconsistencies, missing values, and potential outliers.
3. Handling missing values: I identified missing values in the dataset using 'isnull()' and 'sum()' methods. Depending on the nature of the missing data, I either filled them with the mean or median value using 'fillna()' and 'mean()' or 'median()' methods, or removed rows with missing values using 'dropna()' method to avoid introducing bias or inaccuracies.

Remove private charter, remove number\_flights\_matched != 0

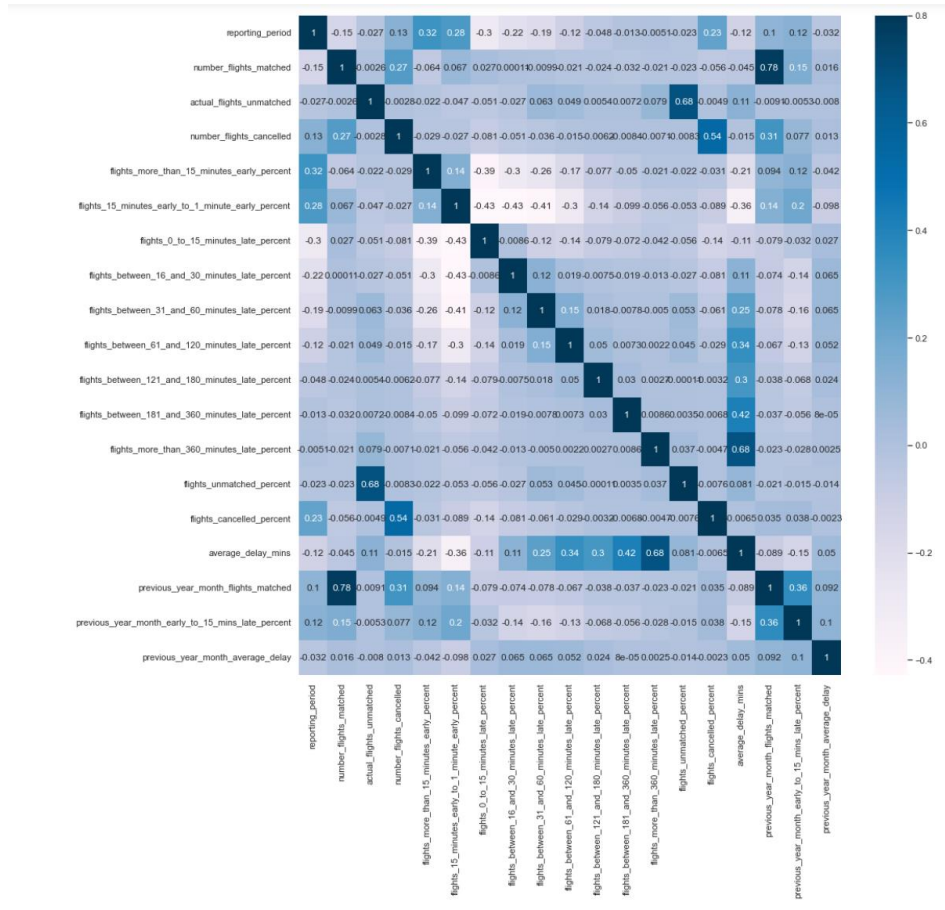
```
In [177]: df_valid = pd.concat([dfs_valid])
          #Remove private charter
          df_valid = df_valid[df_valid["scheduled_charter"]!="S"]
          #Remove empty values
          df_valid = df_valid[df_valid["number_flights_matched"] != 0]
          df_valid
```

Out[177]:

	run_date	reporting_period	reporting_airport	origin_destination	country	origin_destination	airline_name	scheduled_charter	number_flights_matched	ac
0	17/03/2022 17.13	202201	ABERDEEN	HUNGARY	BUDAPEST	WIZZ AIR	S	1		
3	17/03/2022 17.13	202201	ABERDEEN	POLAND	GDANSK	WIZZ AIR	S	8		
4	17/03/2022 17.13	202201	ABERDEEN	UNITED KINGDOM	BELFAST CITY (GEORGE BEST)	LOGANAIR LTD	S	34		
5	17/03/2022 17.13	202201	ABERDEEN	UNITED KINGDOM	BIRMINGHAM	LOGANAIR LTD	S	63		
6	17/03/2022 17.13	202201	ABERDEEN	UNITED KINGDOM	BRISTOL	LOGANAIR LTD	S	40		
...	...	...	...	...	...	...	...	...	...	
2773	28/02/2023 09.17	202212	TEESSIDE INTERNATIONAL AIRPORT	UNITED KINGDOM	ABERDEEN	LOGANAIR LTD	S	69		
2774	28/02/2023 09.17	202212	TEESSIDE INTERNATIONAL AIRPORT	UNITED KINGDOM	BELFAST CITY (GEORGE BEST)	LOGANAIR LTD	S	32		
2782	28/02/2023 09.17	202212	TEESSIDE INTERNATIONAL AIRPORT	IRISH REPUBLIC	DUBLIN	LOGANAIR LTD	S	6		
2783	28/02/2023 09.17	202212	TEESSIDE INTERNATIONAL AIRPORT	NETHERLANDS	AMSTERDAM	KLM	S	104		
2784	28/02/2023 09.17	202212	TEESSIDE INTERNATIONAL AIRPORT	SPAIN	ALICANTE	RYANAIR	S	18		

26842 rows x 25 columns

4. Removing duplicates: I checked for and removed any duplicate records in the dataset using 'duplicated()' and 'drop\_duplicates()' methods to ensure that the data accurately represented unique flights and their associated information.
5. Data type conversion: I made sure that all the features in the dataset had the correct data types using the 'astype()' method. For example, I converted date and time columns to datetime objects using to 'datetime()' method, ensuring that they could be easily manipulated and visualized.
6. Heat mapping of data: Plotting heat map visualisation data. Analysing the image visualises the correlation between the data. The larger the value of the heat map, the higher the correlation between the two data.



## Predict Module

The experimental Integrated Development Environment (IDE) is Jupyter Notebook and Visual Studio Code (VSCode).

In the Predicting Flight Delays module, I utilized the machine learning method that I have implemented in this project to obtain predictive data on flight delays. By leveraging the Light Gradient Boosting Machine (LGBM) algorithm, the goal is to analyse historical flight data and extract patterns that can help predict future delays. This predictive data will serve as a valuable resource for users, enabling them to make informed decisions while choosing airlines and planning their trips. By incorporating this machine learning-based approach into the flight data visualization webpage, we aim to enhance the user experience and provide actionable insights to help travellers minimize disruptions and ensure smoother journeys.

## Train model use lgbmregressor

```
In [13]: 1 import xgboost as xgb
2 model =LGBMRegressor(objective='regression', num_leaves=60, learning_rate=0.01, n_estimators=400,n_jobs=10,metric='mse')
3 model.fit(X_train,y_train,eval_set=[(X_test, y_test)])
4 # model = xgb.XGBRegressor(max_depth = 400, min_child_weight=0.5, subsample = 1, eta = 0.3, num_round = 1000, seed = 1)
5 # model.fit(X_train,y_train)
6 result = model.predict(X_test)
7 MSE = mean_squared_error(result,y_test)
8 testMSE=MSE
9 print("testMSE",testMSE)
10
```

```
[383] valid_0's 12: 1211.41
[384] valid_0's 12: 1211.34
[385] valid_0's 12: 1211.27
[386] valid_0's 12: 1211.09
[387] valid_0's 12: 1211.1
[388] valid_0's 12: 1211.08
[389] valid_0's 12: 1211.09
[390] valid_0's 12: 1211.07
[391] valid_0's 12: 1211.07
[392] valid_0's 12: 1211.03
[393] valid_0's 12: 1210.91
[394] valid_0's 12: 1210.79
[395] valid_0's 12: 1210.71
[396] valid_0's 12: 1210.61
[397] valid_0's 12: 1210.57
[398] valid_0's 12: 1210.61
[399] valid_0's 12: 1210.57
[400] valid_0's 12: 1210.61
testMSE 1210.6142873375723
```

## Predict 2022

```
In [15]: 1 #predict 2022
2 result = model.predict(X_vaild)
3 result
4 df_vaild["predict"] = result
5 df_vaild
```

Out[15]:

## Groupby airline\_name

airline_name	agg_average_delay	predict
(ITA) ITALIA TRASPOTO AEREO	9.688674	9.168854
2 EXCEL AVIATION LTD T/A THE BLADES BROADSWORD SCIMITAR SABRE AN...	0.000000	25.528269
AEGEAN AIRLINES	12.730177	2.836696
AER LINGUS	13.211614	9.864566
AER LINGUS (UK) LTD	18.160269	22.665134
AEROFLOT	10.250654	4.207369
AEROITALIA SRL	46.178572	47.323634
AEROMEXICO	9.546818	7.684121
AIR ALBANIA SHPK	26.793899	17.564547
AIR ALGERIE	30.484733	11.882017

Taking the airlines as a group, calculate an average of the delay time of all months. It can be seen that some of the predicted results and the actual values are relatively close, but some of them are far apart, which may be caused by the relatively small data of some airlines.

Although LGBMs are known for their efficiency in handling large datasets and providing high-quality predictions with low computational requirements, the results in my experiments were not as accurate as expected. There are several reasons why the LGBM algorithm might not work as expected:

1. **Limited amount of data:** Two years of data from 2019 and 2020 were used in my experiments, which resulted in a dataset used to train the model that was too small to capture the underlying patterns and relationships needed for accurate flight delay predictions. In the future, it can be improved by using more years of data, such as using the data of the past ten years to achieve more accurate predictions.
2. **Lack of rich features:** The features used in the dataset may not provide enough information for the algorithm to effectively learn the underlying patterns. Adding more relevant features, such as weather conditions, airport congestion, and seasonality, may improve the model's ability to predict flight delays more accurately. Factors such as the

deviation of passenger flow caused by different airports and holidays are also a major factor affecting the accuracy of flights: in my actual investigation, I found that the higher the passenger load factor (the more people) in a flight, the more time it will take to prepare for the flight. It will also increase (passengers will take longer to board, longer to refuel, and more time to load checked luggage, etc.), these factors will most likely lead to flight delays.

3. **Model complexity:** The LGBM algorithm may not have been optimally configured for this specific problem. The choice of hyperparameters, such as the number of estimators, learning rate, and maximum depth, can significantly impact the model's performance. Tuning these hyperparameters through techniques like grid search or random search could potentially improve the model's accuracy.
4. **Inherent unpredictability:** It is important to note that flight delays can sometimes be caused by unforeseen events, such as sudden weather changes or mechanical issues, which may be difficult to predict even with a well-trained machine learning model.

### Data Visualisation Module

Before visualizing the data, I first conceived the various data that need to be displayed on the page.

1. Total flights for the month.
2. Ranking of airlines in the number of flights.
3. Flight delay rate (by duration of delay).
4. Top 10 flight cancellations.
5. Average delay time (predict value and real value.)
6. Top10 Complaints per million passenger.

In order to display these data, I have selected the appropriate chart type.

1. Circle figure chart
2. Pie Chart
3. Vertical bar chart
4. Horizontal bar chart
5. Line Chart
6. Table chart

Initially, I attempted to create the chart using ECharts. However, I found that the aesthetics and interaction of the whole webpage effects were not satisfactory. After reviewing other people's successful projects, I decided to use a combination of ECharts and Bootstrap frameworks for my project. The visualizations were mainly created using ECharts, while the page elastic layout was implemented using Bootstrap's grid system. Each image has an interactive design. The amount of data can be displayed when the mouse is placed over the icon.

### Filter box

In order to improve the aesthetics of the webpage and facilitate user operations, I added three selectors to the webpage. Users can easily select the date they need to view data according to their own needs. The design of these selectors is simple and easy to use, enabling users to

quickly locate the information they are interested in, thereby improving user experience.

2019-01	2022-01	2019Q1
2019-01	2022-01	2019Q1
2019-02	2022-02	2019Q2
2019-03	2022-03	2019Q3
2019-04	2022-04	2019Q4
2019-05	2022-05	2020Q1
2019-06	2022-06	2020Q2
2019-07	2022-07	2020Q3
2019-08	2022-08	2020Q4
2019-09	2022-09	2021Q1
2019-10	2022-10	2021Q2
2019-11	2022-11	2021Q3
2019-12	2022-12	2021Q4
2020-01		2022Q1
2020-02		2021Q2
2020-03		2021Q3
2020-04		2021Q4
2020-05		2022Q1
2020-06		2021Q2
2020-07		2021Q3
2020-08		2021Q4

### Circle Figure Chart

For displaying the total flights for the month, a circle figure chart was used. This chart type provides an intuitive representation of the total flights by using a circular shape filled with icons or symbols representing flights.

#### Implementation:

To create a circle figure chart using ECharts, I first initialized an ECharts instance with the appropriate container element. I then configured the chart options, such as the series type, symbol type, and data. Finally, I called the 'setOption()' method to render the chart with the

given configuration.

Result:



### **Pie Chart**

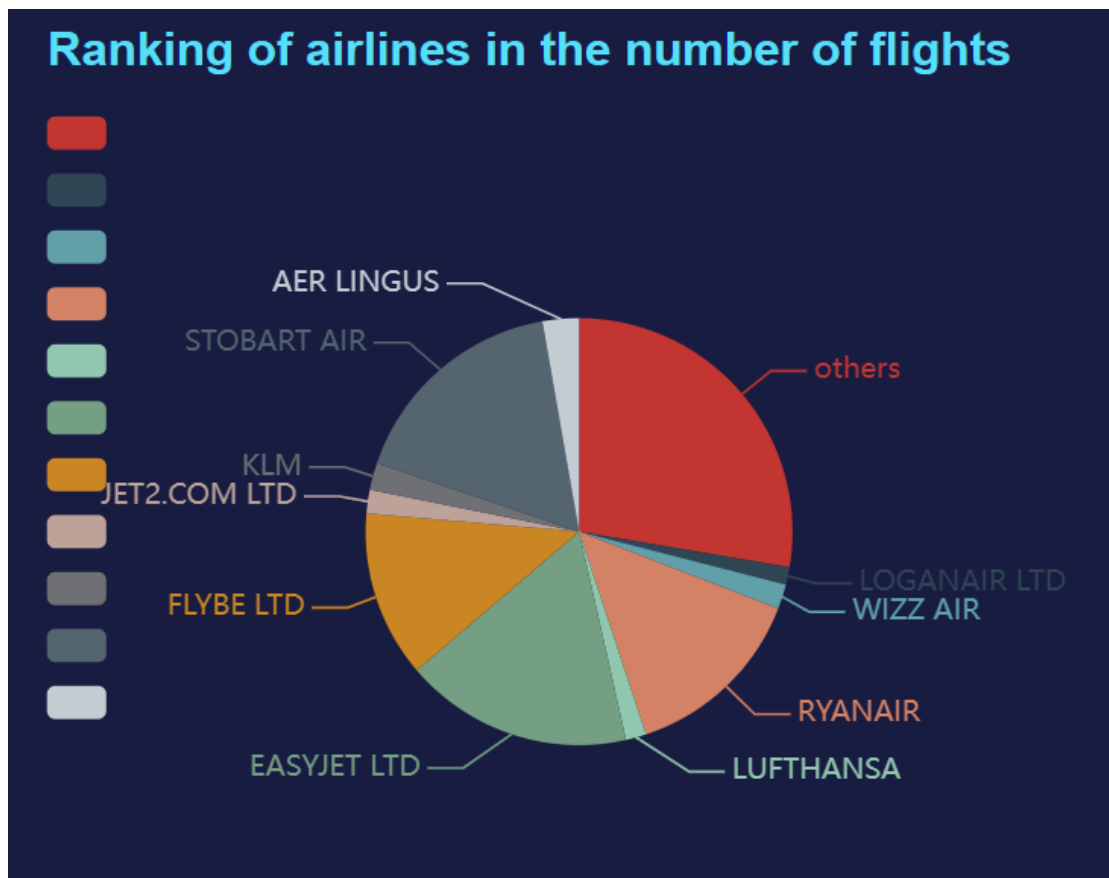
A pie chart was used to display the ranking of airlines by the number of flights. Pie charts are effective in showing proportions of categories within a dataset. Because there are so many airlines. In order to make the pie chart look more intuitive and uncomplicated. I've shown the top ten separately and grouped the others as 'Others' to simplify the chart.

Implementation:

Using ECharts, I created a pie chart by initializing an ECharts instance and configuring the chart options. I set the series type to 'pie', provided the data with labels and values, and configured the visual properties, such as colours and tooltips. After configuring the options, I

used the 'setOption()' method to render the pie chart.

Result:



### Vertical Bar Chart

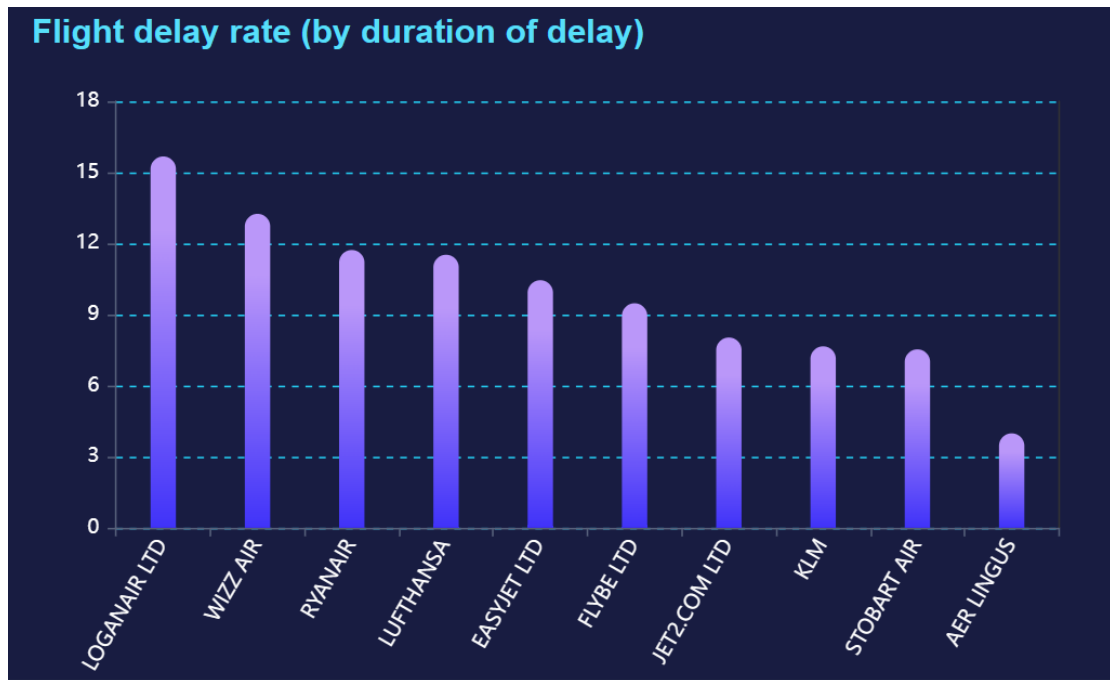
A vertical bar chart was used to display the flight delay rate by the duration of the delay. This type of chart is suitable for comparing categorical data across different groups.

Implementation:

To create a vertical bar chart using ECharts, I first initialized an ECharts instance with the appropriate container element. I configured the chart options, such as the X-Axis and Y-Axis properties, series type, and data. I also customized the visual properties, such as colours, tooltips, and labels. Finally, I called the 'setOption()' method to render the chart with the given

configuration.

Result:



### Horizontal Bar Chart

A horizontal bar chart was used to display the top 10 flight cancellations. This chart type is useful for presenting ranked data and is easier to read when dealing with long category labels.

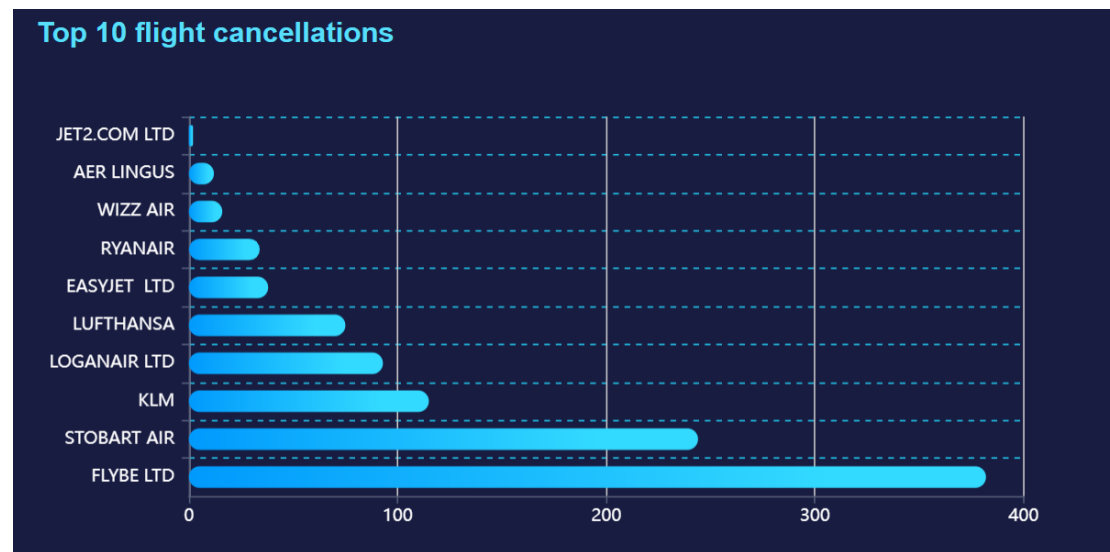
Implementation:

Using ECharts, I created a horizontal bar chart by initializing an ECharts instance and configuring the chart options. I set the series type to 'bar', provided the data with labels and values, and adjusted the X-Axis and Y-Axis properties to display the bars horizontally. I also configured the visual properties, such as colours, tooltips, and labels. After configuring the



options, I used the 'setOption()' method to render the horizontal bar chart.

Result:



### Line Chart

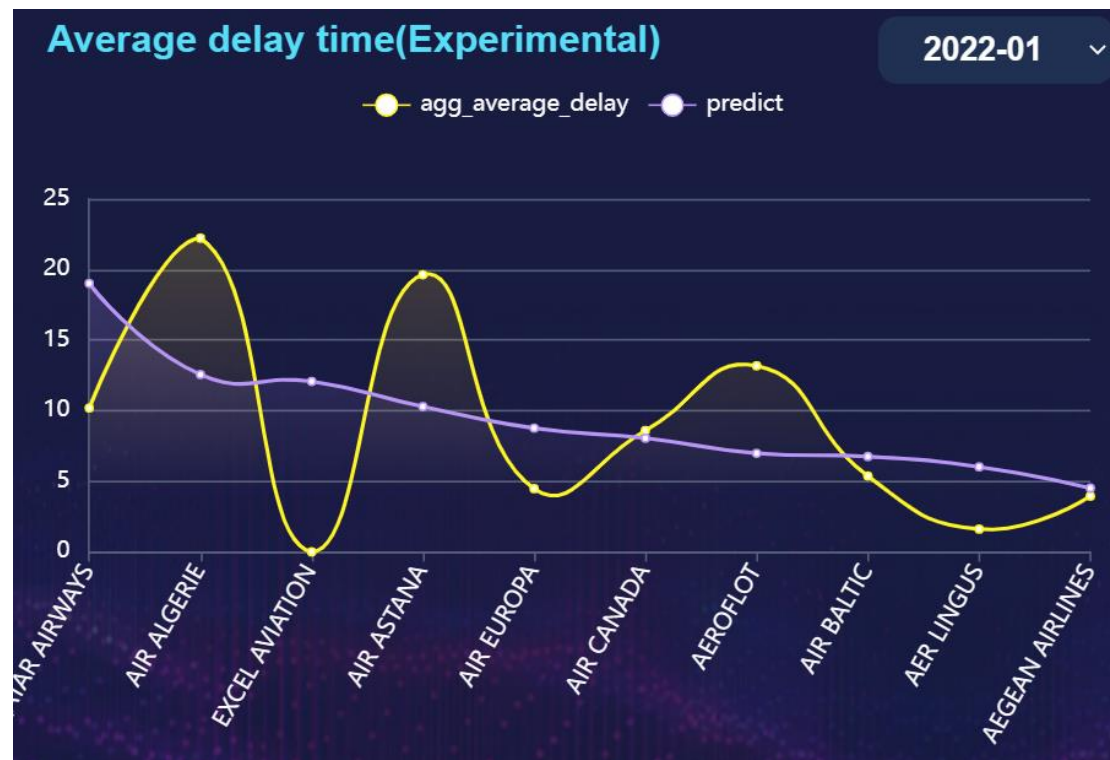
A line chart was used to display the average delay time, comparing predicted values with real values. Line charts are effective in showing trends and patterns over time or across different categories. Using line graphs to show forecast as well as real delay data can show trends in the airline's data. But the end result is a nuisance to the user. It would therefore be more appropriate to change to bar charts or other charts in the future.

Implementation:

To create a line chart using ECharts, I first initialized an ECharts instance with the appropriate container element. I configured the chart options, such as the Axis and Y-Axis properties, series type, and data for both predicted and real values. I customized the visual properties, such as colours, tooltips, and labels, to distinguish between the two lines. Finally, I called the

‘setOption()’ method to render the chart with the given configuration.

Result:



### Table Charts

A table chart was used to display the top 10 complaints per million passengers. Tables are useful for presenting structured data in an organized and easy-to-read format.

Implementation:

To create a table chart, I first used Bootstrap's responsive table component and grid system for the overall elastic layout of the page, ensuring that the table, as well as other elements on the

page, would display correctly on various devices and screen sizes.

Result:



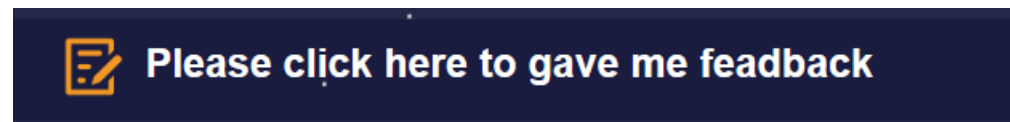
Webpage final result:



## Evaluation & Feedback

Incorporating user feedback in the development process is necessary, as it helps identify areas for improvement, validate the effectiveness of the web application, and ensure it meets users' needs and expectations. This iterative approach allows for continuous refinement and enhancement of the user experience.

To collect user feedback and evaluate the project, I designed a questionnaire and integrated it into the web. The questionnaire was carefully crafted to gather on various aspects of the web, such as the visual appeal, ease of use, relevance of the displayed data, and overall satisfaction.




The questionnaire and responses can be found with the following URL:  
<https://forms.microsoft.com/e/7wSKVMiN6n>

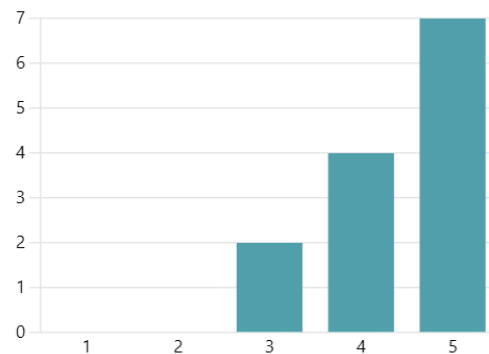
As of April 1, 2023, I have received a total of 15 responses to the questionnaire. Among them, the question "Do you think this webpage is useful to you?" received a rating of 4.38 out of 5. 54% of users rated it a five, 31% rated it a four, and 15% rated it a three. Satisfaction scores in other aspects of the project are also above four. Feedback collected on areas where the webpage could be improved is valuable in determining future directions for the site and identifying necessary enhancements.

1. Do you think this webpage is useful to you?

[More Details](#)

 Insights

4.38  
Average Rating



## Deficiencies of the Project

There were some limitations in the dataset that could have affected the analysis and predictions. One such limitation was the inclusion of data from FlyBe, an airline that went bankrupt during the period covered by the dataset. This oversight may have had several implications on the project:

1. Inaccurate predictions: Since FlyBe ceased operations during the timeframe of the dataset, its data may not accurately reflect current trends in the aviation industry. Including this data in the analysis could have led to inaccurate predictions, particularly when it comes to flight delays.

2. Skewed comparisons: When comparing different airlines, the inclusion of FlyBe data might have skewed the results, making it difficult for users to make informed decisions based on the visualizations.
3. Irrelevant information: Since FlyBe is no longer operational, its data may not be relevant to users looking to choose an airline for their travel needs. Including such data in the visualization could lead to confusion and detract from the user experience.
4. I created a flight visualization data page and initially opted to store the data in a JSON file to expedite the project's progress. However, using a database like PostgreSQL (PSQL) would be a more appropriate choice for storing the data, as it enhances maintainability and allows for seamless modifications and updates.

## Further Work

### **Add an account system.**

Considering the future development of the website, incorporating an account system is crucial. By implementing an account system, users' inquiry history can be tracked and displayed, streamlining the process for users who make multiple inquiries. This approach enhances user experience by providing personalized and easily accessible information, ultimately making the website more efficient and user-friendly.

### **Add the search system.**

Integrating a search system into the platform will significantly enhance user experience and efficiency. The retrieval system enables users to directly search for the specific flight data they require, streamlining the process of data acquisition. By implementing advanced search features such as filters and autocomplete suggestions, users can easily and quickly access relevant flight data tailored to their needs. This addition not only improves the overall functionality of the platform but also encourages user engagement and satisfaction, as they can effortlessly obtain the information they seek.

### **Add the database system.**

Incorporating a seamless integration of front and back ends, along with transitioning from JSON to a database for data storage, will substantially improve the overall performance and efficiency of the platform. By adopting a robust database system, the platform can effectively manage and store large volumes of flight data, facilitating quicker retrieval and more reliable data management. This change will ensure scalability, allowing the system to accommodate future growth in data and user traffic without compromising performance.

## Project Managements

At the beginning of the project, I recognized the importance of effective time management and created a comprehensive project timeline. The timeline outlines the various tasks that need to be completed within a limited amount of time, ensuring that each phase of the project is allocated sufficient time to complete. By setting milestones and deadlines, I aim to maintain a steady pace throughout the duration of the project.

During the course of the project, it became apparent that some tasks required more time than initially expected. Therefore, schedules must be adjusted to accommodate these unforeseen challenges. These adjustments allow me to reallocate time and resources to ensure each part of the project gets the attention it deserves.

In order to stay on track and manage my time effectively, I always monitor my progress against the schedule, note any deviations and take corrective action as needed. This approach helped

me identify potential bottlenecks early on, giving me the opportunity to revise my plans and minimize the impact of any delays on the overall project timeline.

In addition to planning and monitoring, regular communication with mentors played a vital role in the successful management of the program. By keeping all mentors updated on how the project is going and getting feedback, I am able to effectively address any concerns or issues that arise along the way.

# 2022-2023

PROJECT  
XINGYU WANG

PROJECT SCHEDULE	START DATE	DEAD LINE		DEAD LINE
PROJECT PROPOSAL	26/9	14/10	PROJECT DEMONSTRATION	Week8
DATA ACQUISITION AND FILTERING	10/10	21/10	FINAL SUBMISSION	1/11
EXPERIMENT WITH THE DATA	24/10	24/12		
PROJECT INSPECTION	Week7			
VISUALISATION CODE WRITING	1/1	1/3		
PROJECT COMPLETION AND IMPROVEMENT	1/3	10/4		

## Achievements

### Data processing

The use of the pandas library for data processing proved to be a good decision for the project. The library provided a comprehensive set of features that met all data manipulation requirements and simplified the development process. Although there were initial challenges in reading the data, these were eventually resolved through diligent troubleshooting and optimisation. The ability to overcome these obstacles not only demonstrates the flexibility of the pandas library, but also its robustness in handling complex data processing tasks. Mastering the use of the pandas library will be very useful for my future studies as well as my work.

### Choice of technologies and frameworks

The decision to use the JavaScript ecosystem has been a great success as the integration is seamless throughout. An added bonus is that I don't have to spend as much time learning a new language to implement some features as I do improve my skills to use a specific framework. Using ECharts in combination with bootstrap for development has been particularly successful. The results achieved using them combine both aesthetic and interactive design. This makes my pages look perfect.

### Project Management

Taking on a software development project alone for the first time in my three years at university presented me with a unique project management challenge. Balancing the requirements of the project with my other coursework required careful planning and prioritisation.

I encountered difficulties throughout the project, particularly when I realised that ECharts did not meet my aesthetic requirements. This realisation led me to invest a significant amount of

time in redesigning the web pages, which added additional stress to an already tight schedule.

Despite these hurdles, I was able to overcome them through perseverance, effective time management and adaptability. I learned to make better use of my time and resources. This has enabled me not only to meet deadlines but also to gain valuable insights into handling similar challenges in future work.

This experience has taught me the importance of flexibility, continuous learning, and the ability to adapt to unforeseen circumstances. These experiences will undoubtedly serve me well in my future projects and career in software development.

## Conclusion

In summary, this project successfully achieved its main objective of developing a visual data webpage for the dataset provided by CAA, effectively presenting flight delay and complaint data in a more visually appealing and user-friendly manner. Careful selection of technologies, frameworks and software engineering practices contributed significantly to the overall success of the project, providing a valuable tool for users seeking to more easily retrieve and understand the data.

Despite these achievements, a number of limitations have been identified, providing opportunities for further improvement and refinement. To ensure that the system is suitable for deployment in a commercial environment, there is also a need to improve the predictive accuracy of the LGBM model by training it with additional datasets and parameters.

In addition, exploring other forms of data visualisation could greatly enhance the system and provide users with additional insight and analysis. By incorporating a wider range of visualisation techniques, the platform could cater for different user preferences and allow for deeper exploration of the data, ultimately leading to more informed decisions.

Future work may also involve the integration of real-time data updates, enabling users to access the latest information on flight delays and complaints. This will make the platform more valuable to its users as it will provide accurate and up-to-date insights that can inform their decision-making process.

## Code Reference

```
// Code Reference
// This project makes use of the following libraries and frameworks:
// - JavaScript (JS)
// - Cascading Style Sheets (CSS)
// - jQuery (jq)
// - Bootstrap
// - ECharts
//
// The specific configurations, styles, page layouts, and logic code within these components
// have been customized by the developer to meet the unique requirements of the project.
```

## Reference

Endert, Alex, et al. "The state of the art in integrating machine learning into visual analytics." *Computer Graphics Forum*. Vol. 36. No. 8. 2017.

Zámková, Martina, Martin Prokop, and Radek Stolín. "Factors influencing flight delays of a European airline." *Acta Universitatis Agriculturae et Silviculturae Mendelianae*

*Brunensis* 65.5 (2017): 1799-1807.

Carvalho, Leonardo, et al. "On the relevance of data science for flight delay research: a systematic review." *Transport Reviews* 41.4 (2021): 499-528.

Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.

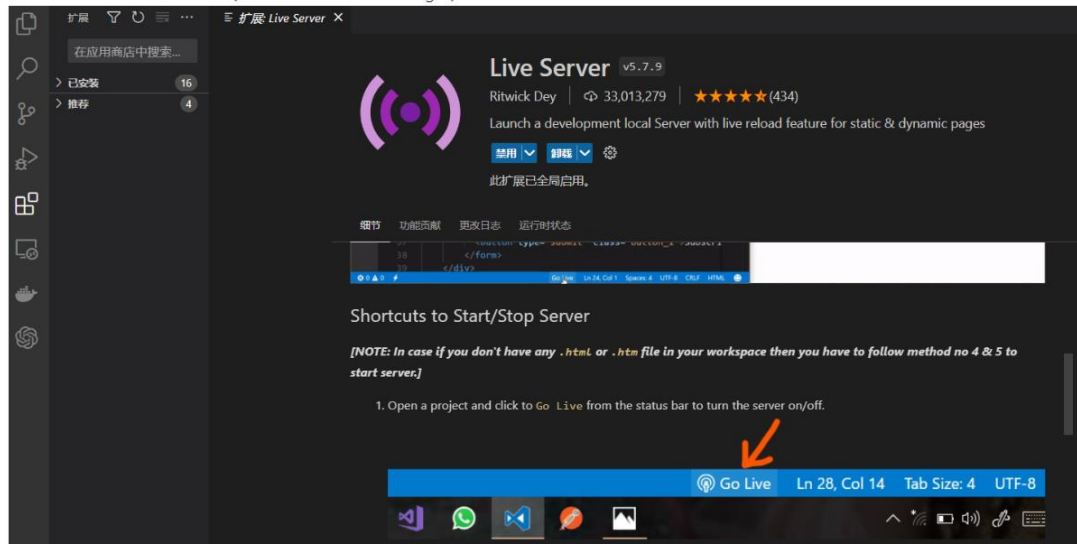


# Appendix

Git Repository: <https://git.cs.bham.ac.uk/projects-2022-23/xxw971.git>

How to open my Project:

1. Install Visual Studio Code (VSCode).
2. Open 'Web' folder in VSCode.
3. Insatll 'Live Server' Extension (Find 'Extension' on the right).



4. Use 'Go Live' button to start web (Should be in the bottom right corner).
5. The Web page should open in your Browser.

'2019Data', '2020Data' are the flight data set downloaded from the CAA.

'Airline Complaints Data', 'Complaint data' are the complaint data downloaded form the CAA.

'Web' is the code of my visualisation website.

'LGBM.ipynb' is my machine learning code written in jupyter notebook.

'predict record.docx' is an experimental record of machine learning.

Name	Last commit	Last update
2019Data	test	2 weeks ago
2020Data	test	2 weeks ago
Airline Complaints Data	test	2 weeks ago
Complaint data	test	2 weeks ago
_MACOSX	test	2 weeks ago
web	Update web/.idea/.gitignore, web/.idea/K...	2 weeks ago
LGBM.ipynb	test	2 weeks ago
README.md	Update README.md	2 days ago
image-1.png	Update README.md, image-1.png	2 days ago
image.png	Update README.md, image.png	2 days ago
predict record.docx	test	2 weeks ago