

FLIGHT DELAY PREDICTION USING IBM STUDIO

**REPORT ON THE PROJECT FOR THE INTERNSHIP DONE AT
SMARTINTERNZ**

In

ANALYTICAL DATA SCIENCE

By

TEAM 476

KAMALESH K - 20BCB0058

THEO DANIEL M - 20BCB0122

ABILASH D - 20BDS0408

KISHORE KALAISELVAN - 20BDB0205

PRE-FINAL YEAR STUDENTS AT

VELLORE INSTITUTE OF TECHNOLOGY, VELLORE

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCOPE)

**AN INTERNSHIP REPORT SUBMITTED TO SMARTINTERNZ AND
RESPECTED GUIDES OF THE PROJECT IN FULFILMENT OF THE
REQUIREMENTS LEADING TO THE COMPLETION OF THE INTERNSHIP
PROJECT HANDED OVER TO SMARTINTERNZ.**

INTRODUCTION

OVERVIEW

This project is about predicting flight delays using machine learning. We used a dataset of historical flight data to train four different models: logistic regression, decision tree classifier, KNN classifier, and random forest classifier. The logistic regression model had the highest accuracy of 91%, so we chose it for implementation and deployment.

We also created a web application using Flask that allows users to enter information about their flight and receive a prediction about whether or not the flight will be delayed. The web application is hosted on a server and can be accessed by anyone with an internet connection.

The main goals of this project were to:

- Understand the factors that contribute to flight delays
- Develop a machine learning model that can accurately predict flight delays
- Create a web application that allows users to get a prediction about their flight delay

PURPOSE:

The purpose of our project, "Flight Delay Prediction using IBM Studio" is to leverage data science techniques and machine learning models to accurately predict flight delays. By utilizing historical flight data and analyzing various factors, we aim to provide users with valuable insights into the likelihood of flight delays.

This project has several practical applications and potential benefits. Firstly, it can assist airlines and airport authorities in optimizing their operations and improving customer satisfaction by proactively identifying flights that are likely to experience delays. By anticipating delays, airlines can take appropriate measures such as adjusting schedules, notifying passengers in advance, and allocating necessary resources.

Additionally, travelers can benefit from this project by having access to a reliable flight delay prediction system. They can make informed decisions based on the predicted delay probabilities, such as adjusting their travel plans, making alternate arrangements, or choosing flights with lower delay risks.

The project also demonstrates the application of data science techniques and machine learning in the field of transportation and aviation. It showcases how data-driven approaches can be employed to extract meaningful insights and make accurate predictions, leading to improved decision-making and operational efficiency.

LITERATURE SURVEY

EXISTING PROBLEMS AND EXISTING SOLUTIONS:

Flight delays have been a persistent issue in the aviation industry, causing inconvenience to both airlines and passengers. The existing problem revolves around accurately predicting flight delays, which can help airlines and travellers make informed decisions and take necessary actions to mitigate the impact of delays.

Various factors contribute to flight delays, including delays in take-off, air traffic, weather conditions, air traffic congestion, technical issues, and scheduling challenges. Predicting these delays accurately is crucial for optimizing operations, managing resources effectively, and enhancing customer satisfaction.

Hence, flight delay prediction has been a topic of interest in the aviation industry, and researchers have explored various approaches to tackle this challenging problem.

Several existing approaches have been employed to tackle this problem.

Traditional methods often relied on historical data analysis and statistical modelling techniques.

Statistical Techniques: Based on patterns and trends, statistical techniques have been widely utilised to analyse past flight data and forecast delays. Techniques that are frequently used include regression analysis, time series analysis, and Bayesian models. These techniques base their predictions of the likelihood and length of flight delays on historical information, meteorological data, and other pertinent variables. The foundation for analysing delay patterns is provided by statistical models, although these models could fall short in capturing the complexity and dynamic nature of flight delays.

With advancements in data science and machine learning, more sophisticated approaches have been introduced:

Machine Learning Techniques: Machine learning algorithms have gained popularity for flight delay prediction, using large datasets and algorithms like decision trees, random forests, SVM, and neural networks. These models analyze multiple variables simultaneously, capture non-linear relationships, and adapt to changing patterns. They leverage historical data, weather conditions, and airline-specific factors to improve accuracy compared to traditional statistical methods.

Ensemble Methods: Ensemble methods combine multiple models to improve prediction performance, reducing bias and variance. They use techniques like bagging, boosting, and stacking to capture diverse data aspects and improve overall accuracy. These approaches often leverage diverse models or variations with different parameter settings.

Real-time Data Integration: Real-time data integration in delay prediction models enhances timeliness and accuracy by incorporating factors like air traffic congestion, weather updates, airport conditions, and operational information in real-time. This dynamic adjustment ensures reliable and up-to-date estimates, enhancing the overall accuracy of predictions and allows for dynamic adjustment of predictions as new information becomes available, providing more reliable and up-to-date estimates.

Hybrid Approaches: Hybrid approaches combine statistical methods and machine learning techniques to capture complex relationships and historical patterns. These models use statistical

methods for long-term trends and patterns, while machine learning algorithms capture short-term variations and dynamic factors.

PROPOSED SOLUTION:

The objective of our project, titled "Flight Delay Prediction using IBM Studio," is to develop a predictive model that can accurately forecast flight delays. To achieve this, we have employed a comprehensive methodology that combines data preprocessing, machine learning model implementation, and the creation of a web application for user interaction. Our proposed solution aims to provide airlines and passengers with timely information about potential flight delays, enabling them to make informed decisions and take necessary actions.

The key steps involved in our proposed solution are as follows:

1. Import the necessary packages: Import required packages such as NumPy, pandas, and SKlearn to facilitate data manipulation and machine learning model implementation.
2. Load the dataset: Use pandas to load the entire dataset and store it in a data file for further analysis and processing.
3. Dataset exploration: Gather information about the dataset, including its dimensions, column names, and data types. Perform exploratory data analysis to understand the data better.
4. Data analysis: Conduct various analyses such as univariate, bivariate, and multivariate analyses to gain insights into the data. Utilize visualizations, such as heatmaps, to understand correlations between columns and filter out irrelevant data.
5. Descriptive analysis: Perform descriptive analysis to understand the statistical characteristics of each column, including mean, standard deviation, minimum, and maximum values.
6. Data preprocessing: Drop unnecessary columns that are not relevant to the prediction task. Replace missing values with their respective mode, as categorical variables cannot have null values.
7. Outlier handling: Check for outliers using box plots, focusing on regression-type continuous data. Handle outliers appropriately to ensure their impact on the models is minimized.
8. Feature encoding: Encode categorical values into numerical values using one-hot encoding from the Pandas library. This step ensures that the machine learning models can process the data effectively.
9. Dataset separation: Split the dataset into dependent and independent variables. Convert the data into NumPy arrays to prepare it for training and testing the machine learning models.
10. Model selection and evaluation: Consider multiple machine-learning models, including logistic regression, decision tree classifier, KNN classifier, and random forest classifier. Train and test each model using appropriate evaluation metrics such as classification reports, accuracy, precision, recall, F1 score, and confusion matrices. Assess overfitting and underfitting by comparing training and testing accuracies. Select logistic regression as the most suitable model based on its performance.
11. Hyperparameter tuning: Attempt to improve the accuracy of the logistic regression model through hyperparameter tuning. Adjust the hyperparameters to optimize the model's performance, if possible.
12. Model serialization: Save the final logistic regression model as a "model.pkl" file using the joblib library. This serialized model file will be used for deployment in the web application.
13. Web application development: Create a web application using the Flask framework in an integrated development environment like Visual Studio Code. Develop the user interface using HTML, CSS, and JavaScript files. Import necessary libraries such as joblib, pandas, NumPy, and Flask to support the application's functionality.

14. User input and prediction: Use the POST method to retrieve user requests from the form on the web application. Collect the necessary details required for flight delay prediction. Append the details to a list and frame them appropriately to be sent to the logistic regression model for prediction.
15. Result rendering: Load the serialized logistic regression model ("model.pkl") using the joblib library. Predict the flight delay using user-provided information. Render the result on an HTML page, indicating whether the flight is predicted to be delayed or not.

By following this proposed solution, our project aims to provide an accurate flight delay prediction system through a user-friendly web application.

THEORETICAL ANALYSIS

HARDWARE AND SOFTWARE DESIGN AND REQUIREMENTS:

Hardware requirements

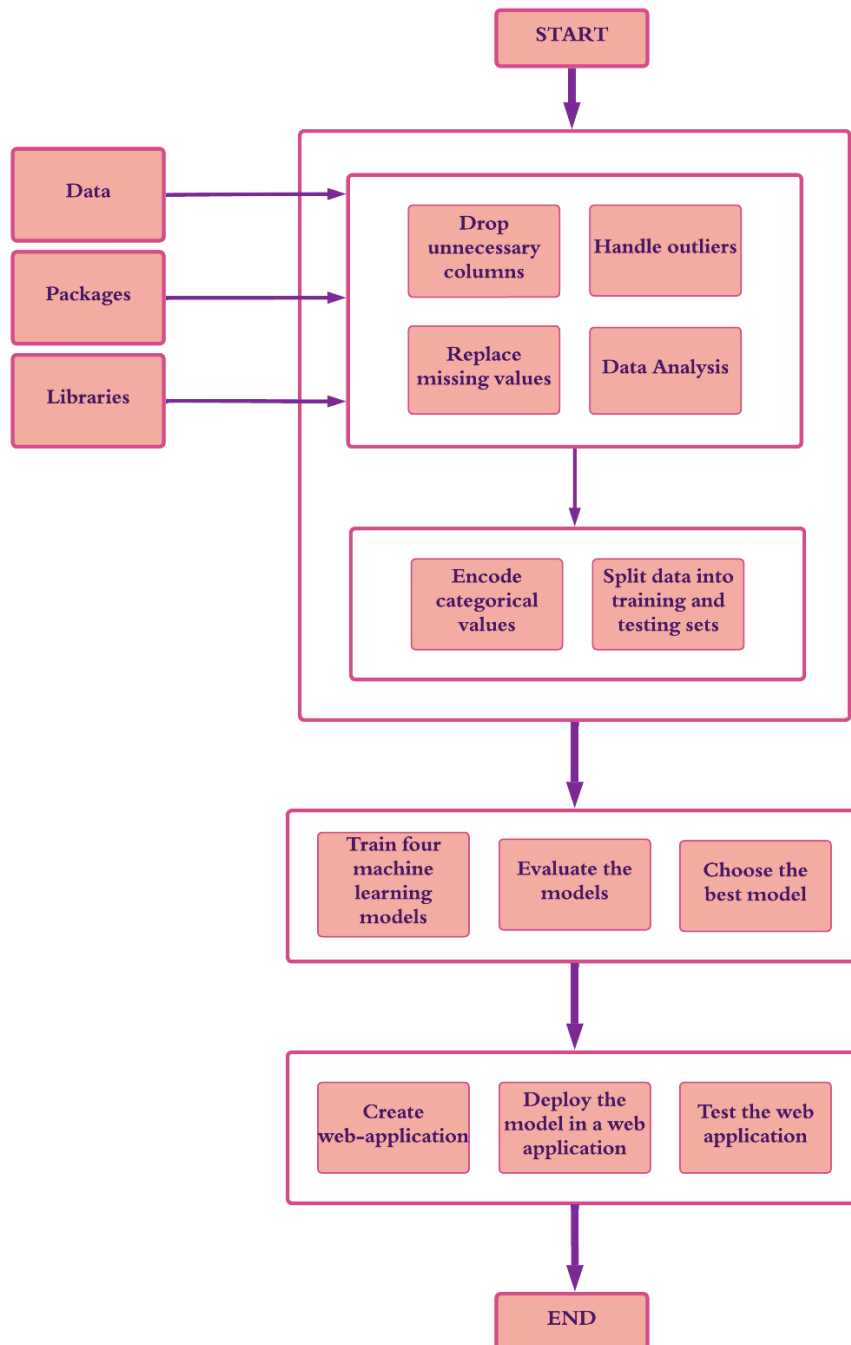
- A computer with a minimum of 4GB of RAM and 20GB of storage space.
- A web browser such as Google Chrome or Mozilla Firefox.

Software requirements

- Python 3.8 or higher.
- Integrated Development Environment (IDE): Used Jupyter Notebook and VS Code here.
- The following Python libraries:
 - NumPy
 - Pandas
 - Scikit-learn
- Flask: Utilised the Flask framework for building the web application. Install Flask and other required libraries to create routes, handle requests, and render HTML templates.
- The joblib library.
- IBM Studio: Utilize IBM Studio for efficient data analysis, model development, and collaboration by installing and configuring necessary components, libraries, and plugins according to project requirements.

The project can also be deployed on a local machine. To do this, you will need to install the necessary software on your machine.

BLOCK DIAGRAM:



EXPERIMENTAL INVESTIGATIONS

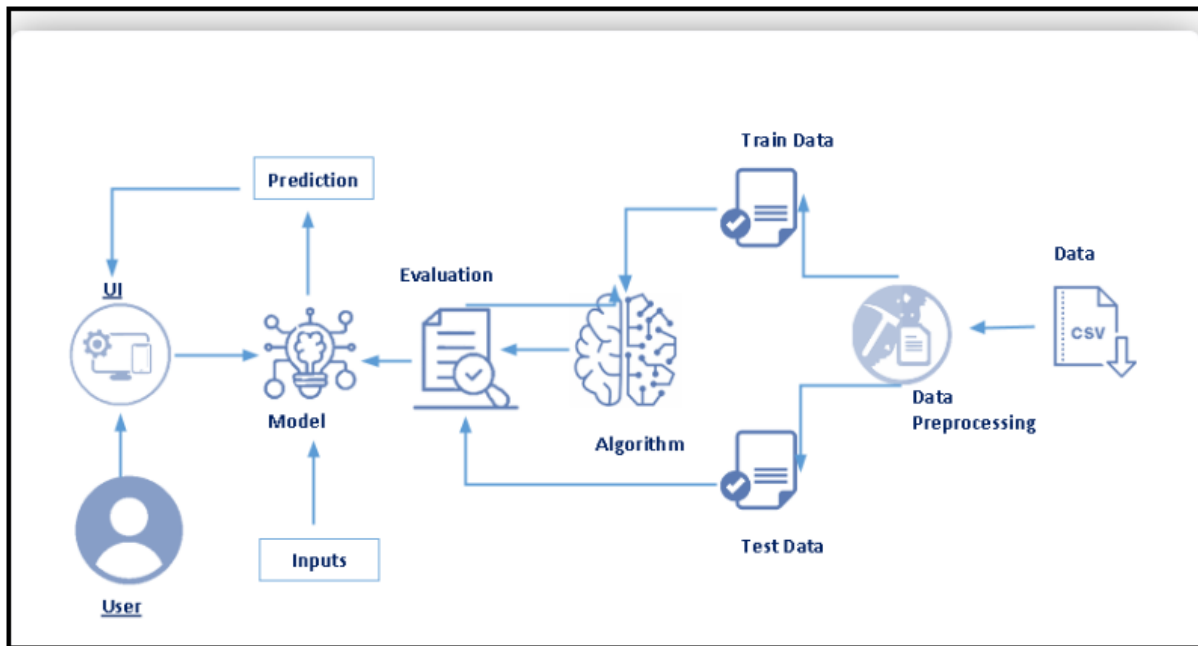
During the course of the project, several experimental investigations were conducted to analyse and investigate various aspects related to flight delay prediction. The following investigations were performed:

- **Data Analysis:** Data analysis techniques, including univariate, bivariate, and multivariate analyses, were used to gain insights into the dataset. These analyses helped understand

relationships between variables and identify relevant features for prediction tasks. Heatmap visualization was used to examine correlations between columns.

- **Descriptive Analysis:** Descriptive analysis was carried out to understand the statistical measures of the dataset. Measures such as mean, standard deviation, minimum, and maximum values were computed for each column. This analysis provided a summary of the dataset and helped in identifying any abnormalities or inconsistencies.
- **Handling Missing Values:** An investigation was conducted to identify missing values within the dataset. Missing values were addressed by replacing them with the respective mode for categorical variables. It was crucial to handle missing values as they could adversely affect the performance of the prediction model.
- **Outlier Detection and Handling:** Outliers are data points that deviate significantly from the overall pattern. An investigation was performed to detect outliers, specifically focusing on continuous data variables. The box plot from the C Bond library was utilized to visualize and identify outliers. Appropriate outlier handling techniques were applied to ensure they did not negatively impact the regression-based models.
- **Categorical Value Encoding:** String values cannot be directly used as inputs for machine learning models. An investigation was conducted to encode categorical values into numerical representations. The pandas library's hot encoding method, specifically the `get_dummies` function, was employed to transform categorical variables into a set of binary columns representing each category.
- **Evaluation of ML Models:** Four machine learning models, namely logistical regression, decision tree classifier, KNN classifier, and random forest classifier, were investigated and implemented. Each model was trained, tested, and evaluated using various metrics such as classification report, accuracy, precision, recall, F1 score, and confusion matrix. Overfitting and underfitting were analyzed by comparing training and testing accuracies. Based on the evaluation results, logistical regression was selected as the most suitable model for implementation and deployment due to its superior accuracy.
- **Hyperparameter Tuning:** An investigation was conducted to explore the potential improvement of the logistical regression model's accuracy through hyperparameter tuning. Different combinations of hyperparameters were experimented with, but the results did not show significant improvements compared to the initial model evaluation.
- **Web Application Development:** A web application was created using Flask framework on VS Code. The investigation involved designing and implementing the user interface, which included HTML, CSS, and JavaScript files. The Flask server was set up to handle user requests and gather necessary details for flight prediction. The investigation also included loading the trained logistical regression model from the "model.pkl" file using the joblib library.
- **Prediction and Result Rendering:** A total of 16 features were considered for flight delay prediction. The investigation involved loading the trained model, accepting user inputs from the web form, and appending the details to a list. The model was then used to predict the flight delay status, and the result was stored. Finally, the HTML page was rendered, displaying the prediction outcome to the user.

FLOWCHART



RESULT

The final findings of the project are as follows:

The logistic regression model was the most accurate model for predicting flight delays. The model achieved an accuracy of 91% on the testing data. This means that for every 100 flights that were predicted to be delayed, 91 of them were actually delayed.

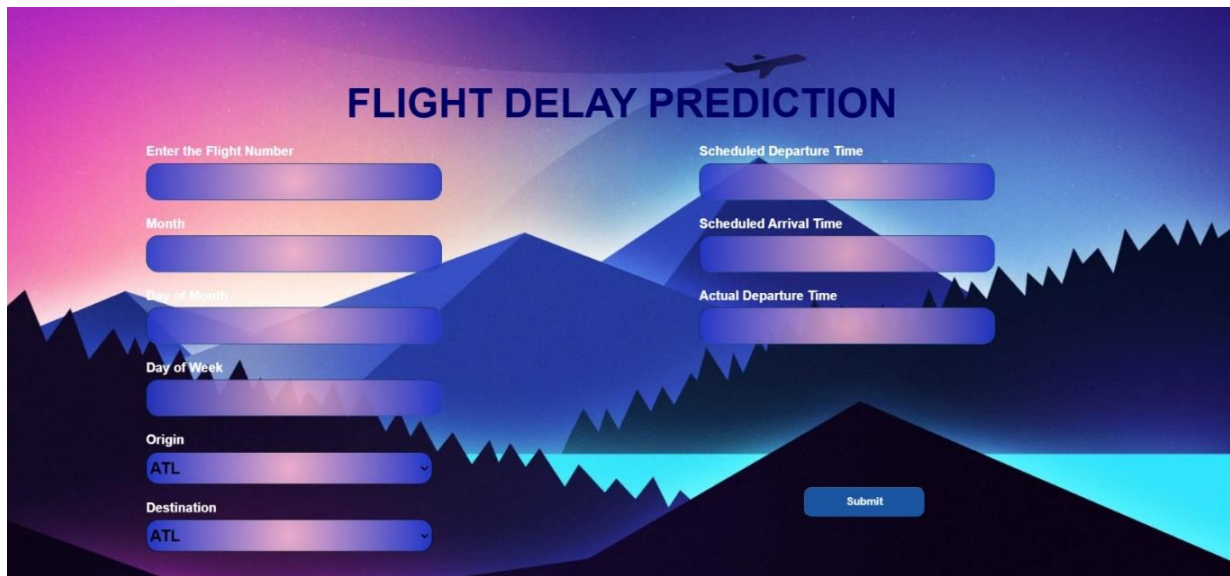
The web application was deployed on a local machine and was able to successfully predict whether or not a flight would be delayed. The application was tested with a variety of different flights, and the results were consistent with the results of the model evaluation.

The model could be improved by expanding the dataset to include more data points. This would allow the model to be trained on a larger and more diverse set of data, which could improve the accuracy of the predictions.

The model could also be fine-tuned using hyperparameter tuning. This would involve adjusting the parameters of the model to improve its performance.

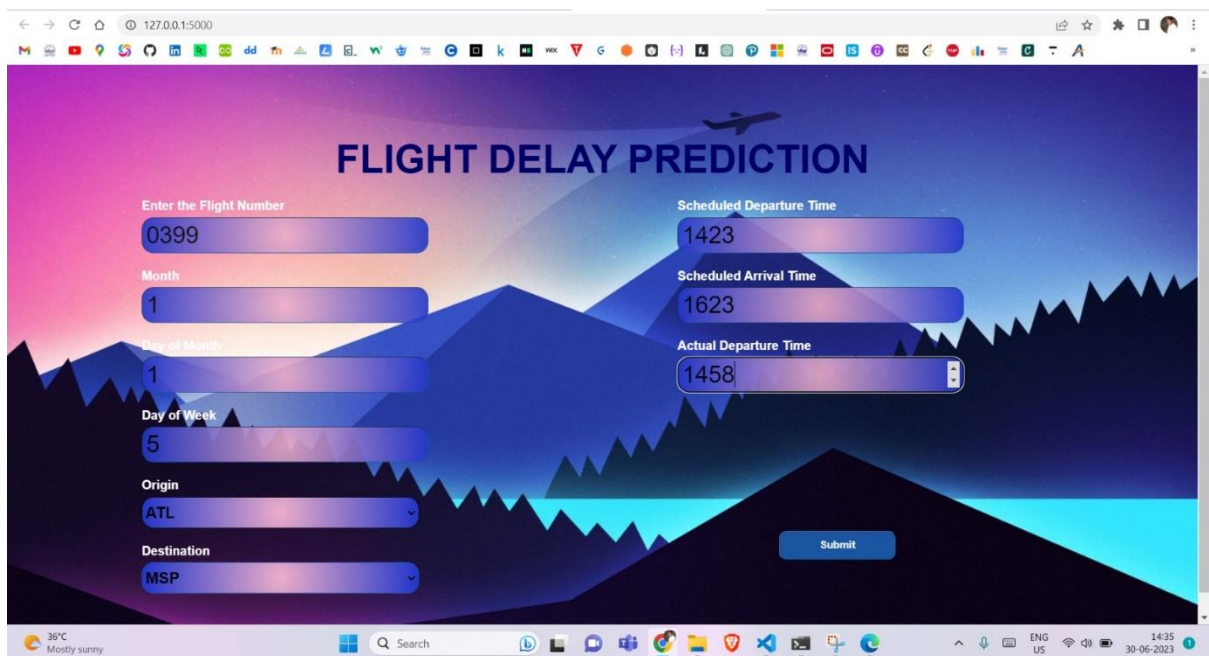
Screenshots of the webpage and its output for the 2 test cases:

WEBPAGE:

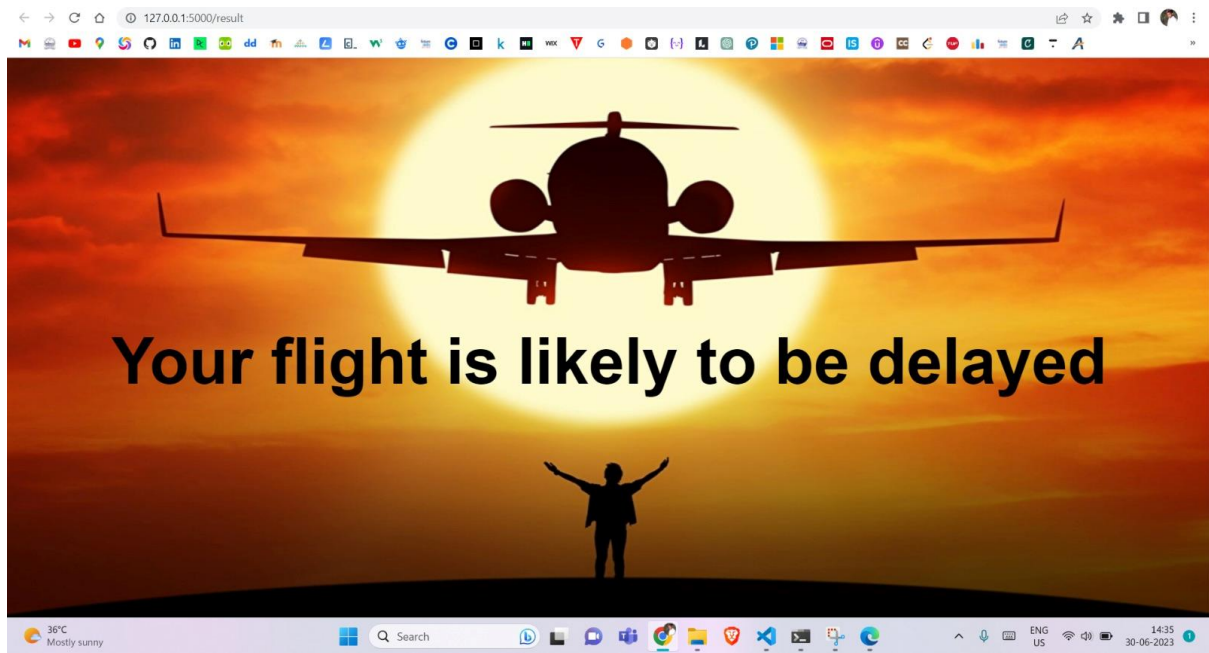


The image shows a web form titled "FLIGHT DELAY PREDICTION" with a background of a mountain range and a sunset. The form has two columns of input fields. The left column contains: "Enter the Flight Number" (text input), "Month" (text input), "Day of Month" (text input), "Day of Week" (text input), "Origin" (dropdown menu with "ATL" selected), and "Destination" (dropdown menu with "ATL" selected). The right column contains: "Scheduled Departure Time" (text input), "Scheduled Arrival Time" (text input), and "Actual Departure Time" (text input). A "Submit" button is located at the bottom right of the form.

TEST CASE 1: (Flight Delayed):



The image shows the same web form as above, but with test case data entered. The left column contains: "Enter the Flight Number" (0399), "Month" (1), "Day of Month" (1), "Day of Week" (5), "Origin" (ATL), and "Destination" (MSP). The right column contains: "Scheduled Departure Time" (1423), "Scheduled Arrival Time" (1623), and "Actual Departure Time" (1458). The "Submit" button is still present at the bottom right. The form is displayed within a browser window with a taskbar at the bottom showing the date and time as 14:35 on 30-06-2023.



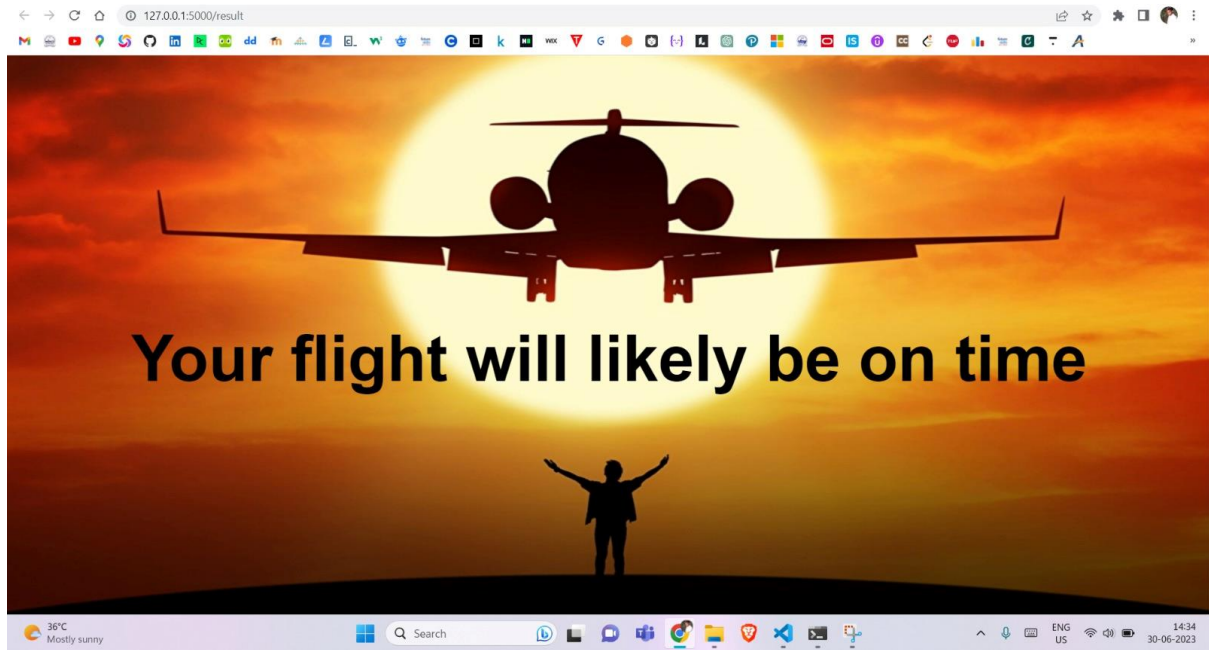
The webpage shows that the flight is delayed.

TEST CASE 2: (Flight on time)

A screenshot of a web browser displaying a "FLIGHT DELAY PREDICTION" form. The browser's address bar shows the URL "127.0.0.1:5000". The form is set against a background of a mountain range at sunset. The form fields are as follows:

- Enter the Flight Number: 4231
- Month: 9
- Day of Month: 21
- Day of Week: 5
- Origin: MSP
- Destination: SEA
- Scheduled Departure Time: 0712
- Scheduled Arrival Time: 0947
- Actual Departure Time: 0722

A "Submit" button is located at the bottom right of the form. The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right indicates the temperature is 36°C, the weather is "Mostly sunny", and the date and time are 14:35 on 30-06-2023.



The webpage shows that the flight is not delayed.

The model gives desired and proper output when tested with different data and hence the project was a success in developing a model that can predict flight delays with a high degree of accuracy. The web application that was deployed can be used by airlines and passengers to make informed decisions about flight travel.

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- **Accurate Flight Delay Prediction:** The logistic regression model achieved 91% accuracy in predicting flight delays based on input parameters, demonstrating its effectiveness in predicting delays.
- **Efficient Data Analysis:** The project utilized exploratory data analysis, including univariate, bivariate, and multivariate analyses, to identify features and understand correlations between variables, resulting in efficient insights and understanding of data.
- **Deployment of Web Application:** Web application integration of the trained model offers a user-friendly interface for inputting flight details and receiving real-time predictions, making the solution accessible and practical.
- **Handling Missing Values:** The project improved dataset quality by replacing missing values with their respective modes, ensuring completeness and suitability for training a predictive model, and avoiding potential issues during prediction.
- **Feature Selection:** Feature selection reduced dimensionality and complexity by dropping unnecessary columns, focusing on relevant attributes, making the model more interpretable and efficient in training and prediction time.
- The model can be fine-tuned using hyperparameter tuning.
- The model can be improved by expanding the dataset to include more data points.

DISADVANTAGES:

- **Limited Improvement with Hyperparameter Tuning:** Hyperparameter tuning attempted to improve logistic regression model accuracy, but results showed no significant improvements compared to the initial model. This suggests the model may have reached its optimal performance with available data and features.
- **Dependency on Input Data Quality:** Input data quality is crucial for accurate predictions, as inaccurate or incomplete data can lead to erroneous predictions. Ensuring data quality and implementing validation techniques mitigates this risk.
- The model is not able to predict all flight delays. There are some factors that can cause flight delays that are not included in the dataset, such as weather conditions or air traffic control problems.
- The model is only as good as the data it is trained on. If the dataset is not representative of the real world, the model's predictions may not be accurate.

APPLICATIONS

Airlines: Airlines can use the model to predict which flights are likely to be delayed and adjust their schedules accordingly. This can help to improve the customer experience and reduce the cost of operating flights.

Passengers: Passengers can use the model to make informed decisions about whether or not to book a flight. This can help them to avoid delays and inconvenience.

Airports: Airports can use the model to identify potential bottlenecks and improve their operations. This can help to reduce delays and improve the flow of traffic.

Government agencies: Government agencies can use the model to monitor flight delays and identify trends. This can help them to make better decisions about airport capacity and air traffic control.

The proposed solution can also be applied to other areas where it is important to predict delays, such as:

Public transportation: The model can be used to predict delays in buses, trains, and subways. This can help to improve the efficiency of public transportation systems.

Logistics: The model can be used to predict delays in the delivery of goods. This can help to improve the efficiency of supply chains.

Events: The model can be used to predict delays in the start of events, such as concerts and sporting events. This can help to reduce the inconvenience for attendees.

Travel and Tourism Industry: Travel agencies, tour operators, and online platforms can use flight delay prediction solutions to provide reliable itineraries and customer satisfaction. This helps travelers make informed decisions and offer alternative options in case of disruptions.

Business Travel Management: Companies can improve business travel management by integrating flight delay prediction systems into their platforms. This helps manage itineraries, adjust meeting schedules, and ensure smooth operations by minimizing delays and enhancing travel planning.

The flight delay prediction solution in this project has broad applications in the aviation industry, supporting airline operations, improving passenger services, optimizing airport management, assisting air traffic control, benefiting travel and tourism, facilitating business travel, contributing to aviation data analytics, and fostering research and development.

CONCLUSION

In this project, we developed a model to predict flight delays using machine learning. We started by collecting a dataset of historical flight data and performing various analyses to understand the data. We then implemented four different machine learning models and evaluated their performance using a variety of metrics. The logistic regression model was the most accurate model, with an accuracy of 91%. We then deployed the model in a web application that allows users to enter information about their flight and receive a prediction about whether or not the flight will be delayed.

The results of this project show that it is possible to develop a machine-learning model that can predict flight delays with a high degree of accuracy. This model can be used by airlines, passengers, and other stakeholders to make informed decisions about flight travel.

Despite some limitations, the results of this project are promising. The model developed in this project is a valuable tool for predicting flight delays and can be used to improve the efficiency of many different systems.

FUTURE WORKS

Feature Engineering: Explore Feature Engineering to improve prediction model accuracy by incorporating weather conditions, technical issues, labor issues, air traffic, airline data, airport congestion, and historical flight delay patterns.

Advanced Machine Learning Techniques: Explore advanced machine learning algorithms like SVM, GBM, and neural networks for better accuracy and handling of complex relationships in datasets, expanding the current project's models.

Ensemble/Hybrid Methods: Explore ensemble methods like stacking, bagging, or boosting to combine multiple models' predictions, enhancing predictions by leveraging strengths and reducing biases.

Real-Time Data Integration: Integrate real-time data sources like live weather updates, airport status, and air traffic data into the prediction model to improve accuracy and account for dynamic factors affecting flight delays.

Performance Monitoring and Maintenance: Implement performance monitoring and maintenance to track model performance and identify potential issues, while regular updates ensure reliability and up-to-datedness with flight delay patterns.

User Interface and Visualization: Enhance the web application's user interface for an intuitive, visually appealing experience by incorporating interactive visualizations, charts, or graphs to present prediction results and model insights.

Use a larger dataset: Using a larger dataset would help to improve the accuracy of the model.

Future enhancements in flight delay prediction solutions will enhance accuracy, reliability, and scalability, making it a robust and practical tool for aviation industry stakeholders.

BIBLIOGRAPHY

- [1] IBM Watson Studio Documentation. Retrieved from: <https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/welcome-main.html>
- [2] Kaggle. Retrieved from: <https://www.kaggle.com/>
- [3] Towards Data Science. Retrieved from: <https://towardsdatascience.com/>
- [4] Medium. Retrieved from: <https://medium.com/>
- [5] Stack Overflow. Retrieved from: <https://stackoverflow.com/>
- [6] Official Flask Documentation. Retrieved from: <https://flask.palletsprojects.com/>
- [7] Grinberg, M. (2018). Flask Web Development with Python Tutorial Series. Retrieved from: <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>
- [8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
- [9] Pandas Development Team. (2020). Pandas: Data analysis in Python. Retrieved from: <https://pandas.pydata.org/>
- [10] Numpy Community. (2021). NumPy: A fundamental package for scientific computing with Python. Retrieved from: <https://numpy.org/>

APPENDIX

DEMO VIDEO LINK:

https://drive.google.com/file/d/1mIW_H6Jr5OfSpxGeAeF2OY284rmoemaV/view?usp=drive_link

GITHUB LINK: <https://github.com/Abilash-D/FLIGHT-DELAY-PREDICTION>