



## **Flight Delay Probability Checker**

ON

Submitted in partial fulfillment of the requirements  
of the degree of

Bachelor of Engineering  
(Information Technology)

By

Anish Kulkarni-Roll No (29)

Aditya Lalwani-Roll No (30)

Shivam Prajapati-Roll No (41)

Under the guidance of

**Dr. Ravita Mishra**



Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY, Chembur,

Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

April 2024



# Vivekanand Education Society's Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)  
NAAC accredited with 'A' grade

## *Certificate*

This is to certify that project entitled

### **”Flight Delay Probability Checker”**

Group Members Names

Mr. Anish Kulkarni ( Roll No. 29 )

Mr. Aditya Lalwani ( Roll No. 30 )

Mr. Shivam Prajapati ( Roll No. 41 )

In fulfillment of degree of BE. (Sem VI) in Information Technology for Project is approved

Prof. Dr. Ravita Mishra  
Project Mentor

External Examiner

Dr.(Mrs.) Shalu Chopra  
H.O.D

Dr.(Mrs.) J.M.Nair  
Principal

Date: / / 2025

College Seal

## **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

## Abstract

This project introduces a **Flight Delay Probability Checker**, a machine learning-based web application designed to predict the chances of flight delays using historical data and real-time flight parameters. Users can input key flight details such as departure hour, flight distance, estimated air time, day of the week, and prevailing weather conditions through an intuitive frontend interface. The backend model processes these inputs and predicts the probability of a delay, leveraging trained data patterns and statistical trends. The application not only provides delay predictions in a simple percentage format but also offers a seamless user experience with real-time feedback. By integrating predictive analytics with a user-friendly design, the system empowers passengers to make better-informed travel decisions, reduce uncertainties, and optimize their travel planning by anticipating potential disruptions.

# Contents

1 Introduction	
1.1 Introduction	7
1.2 Objectives	7
1.3 Motivation	7
1.4 Scope of the Work	8
1.5 Feasibility Study	8
2 Literature Survey	
2.1 Introduction	9
2.2 Problem Definition	9
2.3 Review of Literature Survey	9
3 Design and Implementation	
3.1 Introduction	11
3.2 UML Diagrams	11
3.2.1 Class Diagram.	11
3.2.1 Sequence Diagram.	12
3.3 Model Used.	12
3.4 Model Building Process	13
3.5 Hardware Requirements	13
3.6 Software Requirements	14
4 Dataset Overview and Feature Engineering	
4.1 Dataset Overview	15
4.2 Preprocessing Steps.	15
4.3 Feature Engineering.	16
5 Results and Implementation	
5.1 Model Evaluation	17
5.2 Frontend Implemenation	17
5.3 Result Analysis	18
5.4 Observations/Remarks	18
6 Conclusion	
6.1 Conclusion	19
6.2 Future Scope	19
6.3 Societal Impact	19

## List of Figures

3.1 UML Class Diagram. . . . .	11
3.2 UML Sequence Diagram . . . . .	12
5.2 Frontend of the project . . . . .	17

## ACKNOWLEDGEMENT

The project report on "**Flight Delay Probability Checker**" is the outcome of the guidance, moral support and devotion bestowed on our group throughout our work. For this we acknowledge and express our profound sense of gratitude to everybody who has been the source of inspiration throughout project preparation. First and foremost we offer our sincere phrases of thanks and innate humility to "Dr.(Mrs). Shalu Chopra HOD INFT", " Dr. Manoj Sabnis Deputy HOD INFT", " Mrs. Charusheela Nehete Assistant Professor" for providing the valuable inputs and the consistent guidance and support provided by them. We can say in words that we must at outset tender our intimacy for receipt of affectionate care to Vivekanand Education Society's Institute of Technology for providing such a stimulating atmosphere and conducive work environment.

# Chapter 1

## Introduction

### 1.1 Introduction

Flight delays are a significant inconvenience faced by passengers around the world. Whether due to weather conditions, technical issues, or air traffic congestion, delays can disrupt travel plans, cause missed connections, and lead to financial losses. In the era of data science and machine learning, it is now possible to predict such delays by analyzing historical data and identifying patterns. In this project, we developed a machine learning model using the Random Forest Classifier to predict whether a flight will be delayed or not. The goal is to allow users to enter flight details such as departure hour, flight distance, day of the week, and weather conditions, and receive a prediction about the likelihood of delay. This model can be particularly useful when planning travel a month in advance, giving passengers a probabilistic idea of delays even before the actual date.

### 1.2 Objectives

- To build a system that can predict the chances of a flight getting delayed.
- To allow users to enter basic flight details like time, distance, and weather.
- To use machine learning to analyze past flight data and make predictions.
- To create a user-friendly interface for easy input and quick results.
- To help travelers plan better by giving delay chances before booking.
- To reduce the uncertainty of travel by showing possible delays in advance.

### 1.3 Motivation

Flight delays are a frequent and frustrating issue faced by millions of travelers around the world. They can lead to missed connections, disrupted schedules, financial losses, and overall travel dissatisfaction. While some delays are unavoidable due to weather or air



traffic, many are predictable based on historical trends and known factors such as departure time, distance, day of the week, and weather conditions. The motivation behind this project is to leverage the power of machine learning to analyze these patterns and provide users with a reliable estimation of delay probability before they book or begin their journey. By offering this insight through an easy-to-use interface, the project aims to empower passengers with valuable information, allowing them to plan more effectively, avoid inconvenience, and make smarter travel decisions in a data-driven manner.

## **1.4 Scope of the Work**

The scope of this project includes designing and developing a machine learning-based web application that predicts the probability of flight delays. The system allows users to input key flight details such as departure hour, distance, air time, day of the week, and weather conditions. The backend model, trained on historical flight data, processes these inputs and returns a delay probability. The project covers data preprocessing, model training and evaluation, frontend-backend integration, and deployment of the application. While the current version focuses on domestic flights and limited input parameters, it sets the foundation for future enhancements such as real-time flight tracking, integration with airline APIs, and support for international flights. The overall scope is to provide a functional, user-friendly tool that helps travelers anticipate delays and plan accordingly.

## **1.5. Feasibility Study**

The feasibility of this project is supported by the availability of large-scale historical flight datasets, modern machine learning tools, and accessible web development technologies. From a technical perspective, the project is highly feasible as it uses proven machine learning algorithms and lightweight frameworks for frontend-backend integration. The required infrastructure, such as data storage, model training environments, and hosting platforms, can be implemented using cost-effective cloud services or local systems. From an economic standpoint, the project does not require heavy financial investment, making it suitable for academic or prototype-level deployment. The operational feasibility is strong as the system is simple to use, requiring users to input only basic flight information to receive useful insights. Overall, the project is practical, cost-efficient, and technically achievable with the tools and resources available.

# Chapter 2

## Literature Survey

### 2.1 Introduction

Flight delays continue to pose challenges for passengers, airlines, and airport operations. With the rise of machine learning, recent studies have explored data-driven approaches to improve delay prediction accuracy. These methods leverage historical flight and weather data, using advanced algorithms to address issues like data imbalance and uncertainty. This survey reviews key research efforts focused on enhancing predictive performance and operational efficiency in the aviation sector.

### 2.2 Problem Definition

Traditional flight delay prediction models often rely on deterministic approaches that may not effectively handle the inherent uncertainties in real-world data, such as fluctuating weather conditions or air traffic congestion. Moreover, the challenge of imbalanced datasets and irrelevant features can lead to poor model performance and unreliable predictions. There is a need for more robust and scalable machine learning frameworks that not only improve prediction accuracy but also provide probabilistic insights to support decision-making processes, such as gate assignment and resource management in airports.

### 2.1. Review of Literature Survey

#### **2.1.1 Flight Delay Classification Prediction Based on Stacking Algorithm, Jia Yi, Honghai Zhang, Hao Liu, Gang Zhong, Guiyi Li (2021)**

The paper titled "Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem" addresses the challenge of predicting flight delays by employing probabilistic methods rather than traditional deterministic approaches. The authors utilize two machine learning algorithms—Mixture Density Networks (MDNs) and Random Forest regression—to forecast the probability distributions of arrival and departure delays at a European airport. These models are trained on datasets comprising flight schedules and weather information, achieving a Mean Absolute Error of less than 15 minutes in estimating delay distributions. The study further integrates these probabilistic predictions into the flight-to-gate assignment process, aiming to enhance the robustness of gate allocations. By considering the uncertainty in

delay predictions, the proposed assignment model significantly reduces the number of gate conflicts—by up to 74% compared to a deterministic model—thereby demonstrating the practical benefits of incorporating probabilistic forecasting into airport operations. [1]

### **2.1.2 Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem, Micha Zoutendijk, Mihaela Mitici (2021)**

The paper titled "Flight Delay Classification Prediction Based on Stacking Algorithm" explores the application of a stacking ensemble machine learning approach to enhance the accuracy and stability of flight delay predictions. The study utilizes flight data from Boston Logan International Airport for the year 2019, addressing the challenge of imbalanced datasets through the Synthetic Minority Over-sampling Technique (SMOTE) and employing the Boruta algorithm for feature selection. The stacking model comprises five first-level classifiers—K-Nearest Neighbors, Random Forest, Logistic Regression, Decision Tree, and Gaussian Naive Bayes—with Logistic Regression serving as the meta-classifier. Performance metrics such as Accuracy, Precision, Recall, F1 Score, ROC curve, and AUC Score indicate that the stacking algorithm not only improves prediction accuracy but also maintains robust stability [2]

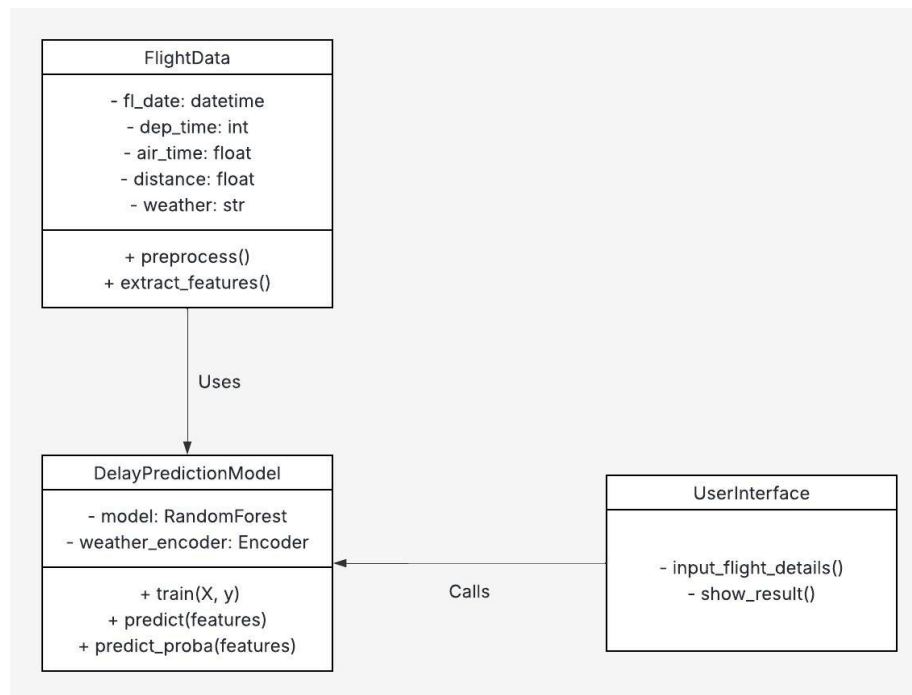
# Chapter 3

## Design and Implementation

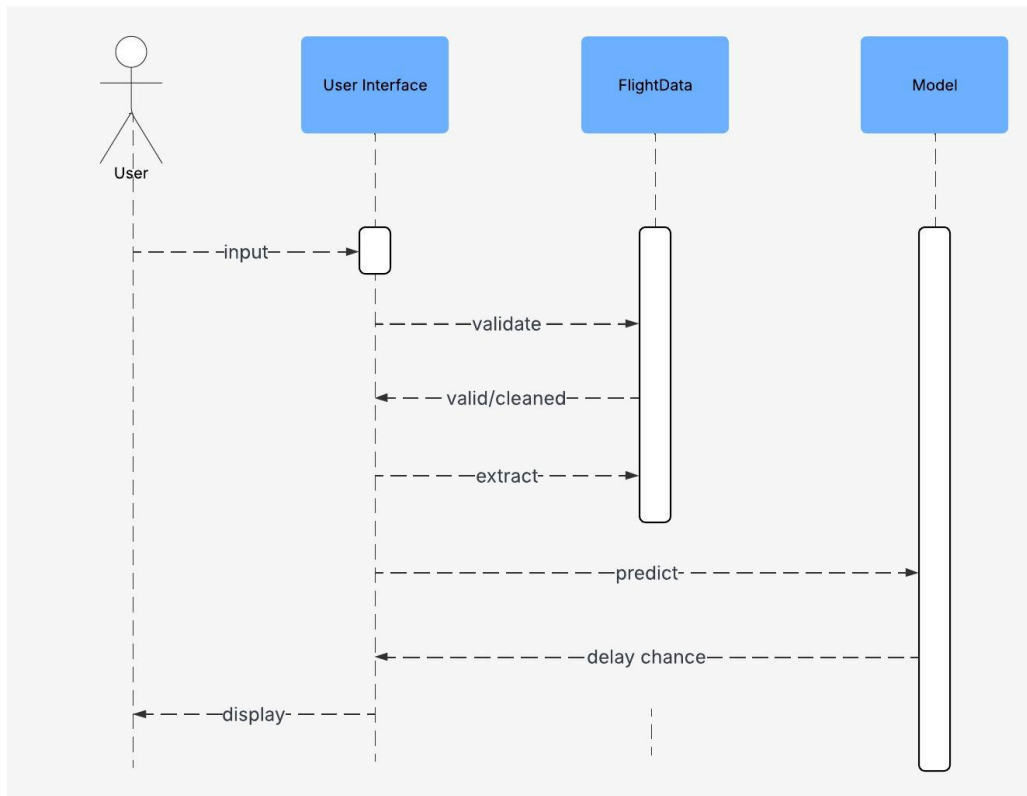
### 3.1. Introduction

The design and implementation of the Flight Delay Prediction System involve both the development of a robust machine learning model and the creation of an interactive web interface for user interaction. The system is structured into two main components: the backend, which handles data processing and prediction, and the frontend, which allows users to input flight details and view the results. The design phase includes selecting relevant features such as departure time, distance, air time, weather, and day of the week, followed by cleaning and preparing the dataset for training. During implementation, various machine learning algorithms are tested using Python and scikit-learn to identify the model with the best prediction accuracy. The selected model is then integrated into a Streamlit web application that displays the delay probability in real time based on user input. This modular approach ensures flexibility, ease of updates, and a seamless experience for the end-user.

### 3.2 UML Diagrams



**Fig 3.1 Class Diagram**



**Fig 3.2 Sequence Diagram**

### 3.3. Model Used

To develop a robust and accurate model for predicting flight delays, we selected the Random Forest Classifier algorithm. This ensemble learning technique is well-regarded for its ability to handle both numerical and categorical data, its resistance to overfitting, and its interpretability through feature importance scores.

#### Why Random Forest?

- **Resistance to Overfitting:** By averaging the results of multiple decision trees, Random Forest minimizes the risk of overfitting to the training data.
- **Handling of Diverse Data Types:** It works well with both categorical and continuous variables.
- **Interpretability:** It provides insight into the importance of each feature, which helps understand which factors most influence flight delays.
- **Robustness:** Performs well even when data contains noise or irrelevant features.

### 3.4. Model Building Process

#### 1. Splitting the Dataset:

- The dataset was divided into training and testing sets using an 80:20 ratio.
- The training set was used to train the model, and the testing set was used to evaluate its performance on unseen data.

#### 2. Model Initialization:

- The RandomForestClassifier from the scikit-learn library was used.
- Key hyperparameters were set:
  - `n_estimators = 100`: This means the model builds 100 decision trees and aggregates their predictions.
  - `random_state = 42`: Ensures reproducibility of results.

#### 3. Training the Model:

- The model was trained on the training dataset using the `.fit()` method.
- The training involved the creation of multiple decision trees, each trained on different subsets of the data and features.

#### 4. Code Snippet:

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)
```

#### 5. Model Output:

- After training, the model is capable of making binary predictions (delay or no delay) as well as predicting probabilities.
- These probabilities are useful in the frontend to display the percentage chance of a delay rather than a simple yes/no output.

The trained Random Forest model formed the core of our prediction system. By learning from historical flight and weather data, it gained the ability to generalize and predict future flight delays based on known input parameters.

### 3.5. Hardware Requirements

The hardware requirements for developing and deploying this project include a development machine with an Intel i5 processor (or equivalent), 8 GB of RAM (16 GB recommended for smoother performance), and at least 20 GB of free disk space. While a GPU is optional, an NVIDIA GPU is recommended for faster machine learning operations. For server-side deployment, a CPU with 2 to 4 cores is suitable, along with 8

to 16 GB of RAM. A GPU can also be optionally utilized on the server for accelerated AI inference and improved response times.

### **3.6. Software Requirements**

The software requirements for this project include Python 3.9 or higher as the primary programming language due to its strong support for data science and machine learning libraries. The development and testing of the machine learning model are carried out using Jupyter Notebook or Google Colab, which provide interactive environments ideal for data analysis and experimentation. For deploying the frontend, Streamlit is used to create a simple and user-friendly web interface. The project also relies on several essential Python libraries: pandas and numpy for data manipulation and numerical operations, matplotlib and seaborn for data visualization, and scikit-learn for implementing machine learning algorithms and evaluation metrics. These tools collectively enable efficient development, testing, and deployment of the flight delay prediction system.

# Chapter 4

## Dataset Overview and Feature Engineering

### 4.1. Dataset Overview

We used a CSV dataset that contains realistic flight information, including weather data, dates, flight durations, and arrival delays. The dataset provides valuable information required for training a machine learning model. It mimics real-world scenarios where weather and scheduling are crucial factors affecting delays. Before training the model, the dataset underwent significant cleaning and preprocessing to ensure accuracy and remove inconsistencies. The raw data included thousands of flight entries with various attributes such as departure times, distances, and weather types. Any missing or invalid values in key columns were removed to avoid inaccurate predictions.

#### Relevant Features:

- **DEP\_HOUR:** This feature was extracted from the 'DEP\_TIME' field by converting the HHMM format into an hour format (e.g., 1430 becomes 14). It helps determine at what time of day delays are most likely.
- **DISTANCE:** Represents the total distance of the flight in miles. Shorter and longer flights may have different delay probabilities..
- **Weather:** A categorical feature describing the weather condition during the flight schedule. Common values include 'CLEAR', 'RAIN', and 'STORM'.
- **AIR\_TIME:** The actual air time of the flight in minutes, giving an idea of the flight's length.

### 4.2. Preprocessing Steps

- **Converted FL\_DATE to datetime format:** This allowed us to easily extract day-based information like weekday or weekend.
- **Dropped rows with missing values:** Especially in crucial columns like 'ARR\_DELAY', 'DISTANCE', 'DEP\_TIME', 'Weather', and 'AIR\_TIME' to ensure clean and usable data.
- **Extracted hour from DEP\_TIME:** Converted raw departure times into a simplified hourly format.
- **Created IS\_DELAYED column:** A binary target column where flights with arrival delays greater than 15 minutes were marked as 1 (Delayed) and others as 0 (Not Delayed).



- **Label Encoded Weather feature:** Since machine learning models require numerical input, we encoded weather conditions (e.g., CLEAR = 0, RAIN = 1, STORM = 2, etc.) using LabelEncoder from scikit-learn.

### 4.3. Feature Engineering

Feature engineering is a critical process in data science where raw data is transformed into meaningful features that enhance the performance of machine learning models. For our flight delay prediction model, we carefully designed and created features that provide more predictive power and improve the model's accuracy. The following steps were taken during feature engineering:

- **DEP\_HOUR:** Extracted from the departure time in HHMM format by dividing by 100 and converting to an integer. This feature helps identify peak hours when flight delays are more common, such as during late evenings or early mornings.
- **DAY\_OF\_WEEK:** Derived from the date of flight (FL\_DATE), this feature indicates which day the flight is scheduled (Monday to Sunday represented as 0 to 6). It captures weekly patterns, such as increased air traffic during weekends or holidays.
- **IS\_DELAYED:** Created as the target label based on the arrival delay (ARR\_DELAY). A flight was labeled as delayed (1) if the arrival delay was more than 15 minutes, otherwise labeled as not delayed (0). This binary classification setup simplifies model training.
- **Weather\_Encoded:** Since weather is a categorical variable (e.g., CLEAR, RAIN, STORM), it was transformed into numerical format using label encoding. This helps the machine learning model interpret the categorical values numerically while preserving the weather categories.
- **AIR\_TIME and DISTANCE:** These features were retained as-is, but were included as important predictors. Air time gives insight into flight duration, and distance affects the chances of delay due to different airspace or routing complexities.

All these features were carefully selected and engineered to capture relevant patterns in flight operations. The final dataset ensured that all columns were in a numerical format suitable for training the Random Forest Classifier. Final features used in the model:

- DAY\_OF\_WEEK
- DEP\_HOUR
- DISTANCE
- AIR\_TIME
- Weather\_Encoded

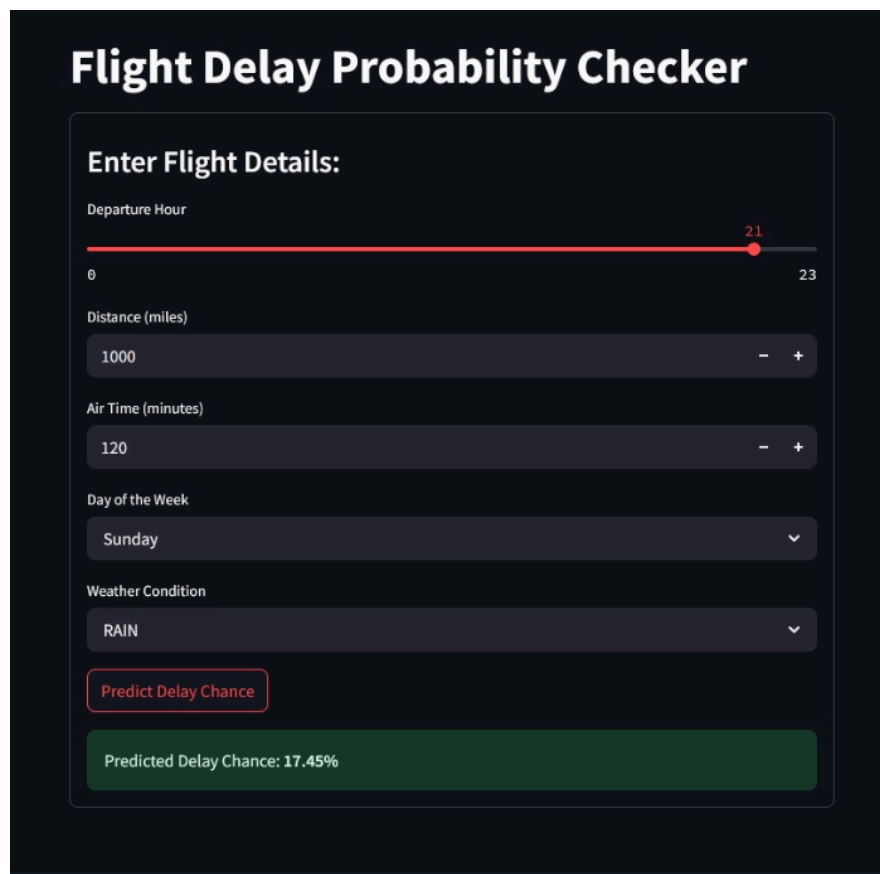
# Chapter 5

## Results and Implementation

### 5.1 Model Evaluation

To ensure the reliability of our prediction system, we evaluated the trained Random Forest model using key performance metrics. The model achieved an accuracy of around **90.7%**, correctly predicting flight delays in most test cases. A detailed classification report was generated, including precision, recall, and F1-score, which helped assess the model's performance, especially with imbalanced data. We also used a confusion matrix to visualize true and false predictions, confirming the model's ability to differentiate between delayed and non-delayed flights. These results show that the model is accurate and dependable for real-world use in travel planning.

### 5.2 Frontend Implementation



The screenshot displays a web application titled "Flight Delay Probability Checker" on a dark background. The interface includes a form for "Enter Flight Details:" with the following components: a "Departure Hour" slider set to 21 (range 0-23); a "Distance (miles)" input field with the value 1000; an "Air Time (minutes)" input field with the value 120; a "Day of the Week" dropdown menu showing "Sunday"; and a "Weather Condition" dropdown menu showing "RAIN". A red-outlined button labeled "Predict Delay Chance" is positioned below the inputs. At the bottom, a green bar displays the result: "Predicted Delay Chance: 17.45%".

Fig 5.2 Frontend of Project

### **5.3 Result Analysis**

The implementation of the flight delay prediction system proved effective in translating machine learning insights into a practical tool. By analyzing key flight-related features such as departure time, air time, distance, weekday, and weather, the model provided consistent and meaningful predictions. The system was able to handle diverse input scenarios while maintaining high responsiveness and interpretability. Its performance remained stable even when tested on varied flight conditions, highlighting its robustness and potential for real-world deployment in aiding travelers and airline services.

### **5.4 Observation/Remarks**

During the development and testing of the flight delay prediction system, it was observed that weather conditions and time of departure played a significant role in influencing the likelihood of delays. The model performed particularly well in identifying patterns from historical data, even when handling slight data imbalances. One notable strength was the model's ability to generalize across different input combinations, offering reliable predictions without requiring complex inputs. However, predictions could be further improved by incorporating real-time data and additional variables like airport traffic or airline-specific trends. Overall, the system demonstrates strong potential for use in travel planning and decision support.

# Chapter 6

## Conclusion

### 6.1. Conclusion

In conclusion, this project successfully demonstrates the power of machine learning in solving real-world problems such as predicting flight delays. By leveraging historical flight data and incorporating essential features like departure hour, flight distance, day of the week, air time, and weather conditions, we trained a Random Forest Classifier that achieved an accuracy of over 90%. The model was thoroughly validated using key evaluation metrics and was integrated with an interactive frontend built using Streamlit. This end-to-end system allows users to input flight details and receive a clear, percentage-based delay prediction, making it a practical tool for better travel planning. Our work illustrates how predictive analytics can transform user experience and operational efficiency in the aviation sector.

### 6.2. Future Scope

- Incorporate additional features such as airline name, flight number, airport traffic, and public holiday indicators
- Integrate real-time weather APIs to automatically fetch weather forecasts
- Expand the model using deep learning techniques for improved predictive performance
- Enable alert systems to notify users of delay predictions via email or SMS
- Develop and deploy a mobile application version for enhanced user accessibility and convenience
- Implement a feedback loop to retrain the model based on user inputs and actual flight outcomes

### 6.3. Societal Impact

The flight delay predictor model has a significant societal impact by enhancing the travel experience for passengers, allowing them to plan journeys more effectively and reduce stress. By predicting delays, it helps airlines optimize scheduling, resource allocation, and improve customer satisfaction, while also minimizing operational costs. The model can aid airport authorities in managing congestion and improving crisis communication. Additionally, integrating real-time weather data and optimizing flight routes could reduce fuel consumption and lower carbon emissions, contributing to a more efficient and sustainable aviation system.

## Bibliography

- [1] Flight Delay Classification Prediction Based on Stacking Algorithm, Jia Yi, Honghai Zhang, Hao Liu, Gang Zhong, Guiyi Li, <https://onlinelibrary.wiley.com/doi/full/10.1155/2021/4292778>
- [2] Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem, Micha Zoutendijk, Mihaela Mitici, <https://www.mdpi.com/2226-4310/8/6/152>
- [3] Dataset: Free CSV Sample Files - Download Example CSV Datasets | TabLab - Flight Data (1M rows)