

Flight Delays Prediction Using Machine Learning : Project Ideation

Background and Motivation :

In today's fast-paced world, flight delays remain as a persistent issue in the aviation industry, causing inconvenience to passengers and financial losses to airlines. With the help of our website, it is possible to predict flight delays with high accuracy and such predictions can empower both passengers and airlines to make informed decisions. Our project aims to develop a machine learning-based system that predicts whether a flight is likely to be delayed, using real-time and historical flight parameters.

Primary Goal :

The main objective is to build a Flight Delay Prediction web application that allows users to input flight details and receive a delay prediction. This includes:

1. Predicting on-time or delayed status based on given inputs.
2. Providing an easy-to-use web interface built with Flask.
3. Ensuring reliable predictions using a trained machine learning model.

These requirements guided our ideation and development process in the initial phase.

Initial Approach and Challenges :

We began our project by taking given datasets related to flight records. After preprocessing and feature engineering, we used Logistic Regression, Random Forest Classifier and Decision Tree Classifier for training.

Key challenges :

1. Data Imbalance: Delayed flights were fewer than on-time flights.
2. Handling Categorical Inputs: Features like Origin and Destination needed one-hot encoding.

Target Users :

1. Frequent flyers and passengers.
2. Airline operation managers.
3. Travel agents.

Tools and Technologies :

1. Python basic libraries such as numpy,pandas,sklearn,matplotlib,seaborn.
2. Flask (for the web interface).
3. Render (for deployment).
4. HTML (for UI).
5. Pickle (for saving ML model).

Problem Statement :

Flight delays disrupt travel plans and cause passenger dissatisfaction.Current systems offer limited real-time predictive insights and there is a need for an intelligent system that can analyze flight details and predict delays effectively.Our project resolves this issue using machine learning to provide accurate flight delay predictions via a user-friendly web interface.

Flight Delays Prediction Using Machine Learning : Requirement Analysis

Functional Requirements :

1. The system must allow users to input flight details (like Flight number,origin, destination,date,day of week,scheduled/actual times).
2. The backend must process this input and return a prediction (On-Time or Delayed).
3. The machine learning model must be pre-trained and integrated using flight.pkl.
4. The prediction output must be shown clearly on the frontend.
5. The system must handle edge cases (e.g., same origin & destination) gracefully.

Non-Functional Requirements :

1. The application should respond within 2–3 seconds.
2. The interface should work on both desktop and mobile browsers.
3. The model and backend should handle unexpected inputs without crashing.
4. The model file (flight.pkl) should be securely stored on the server.

Tools Used :

1. Git : Version control.
2. Render : For Flask app deployment.

3. Numpy,Pandas,Scikit-learn... : For training and inference.
4. HTML : Frontend interface.
5. Jupyter Notebook : For exploratory data analysis and model training.

Code Overview :

Frontend (HTML/JS) :

A form captures inputs like flight date,airports,times and weekday.
JavaScript is used to send POST requests to the Flask API via fetch().

Backend (Python Flask) :

It receives input at predict-delay endpoint.It loads and applies the trained model (flight.pkl) and then preprocesses data using same encoding logic as during training.Finally,it returns prediction in form of "Flight is likely to be delayed" .

Flight Delays Prediction Using Machine Learning : Project Design

System Architecture :

1. Frontend: We built our webpages using HTML pages as it provides a clean user interface for entering flight details such as date, origin, destination and times.
2. Input Handler(JS Logic): JavaScript validates and formats the input and then sends it to the backend via a POST request.
3. Backend(Flask App): The Python-based Flask server (app.py) receives user input, loads the trained ML model (flight.pkl), preprocesses input and predicts the delay.
4. ML Model: A trained machine learning classifier (e.g., Random Forest or Logistic Regression) predicts whether the flight will be delayed or on time.
5. Deployment: The entire application is deployed using Render(for backend) and can optionally be paired with GitHub Pages for the frontend.

We identified two use cases:

1. Passenger Use Case: A frequent flyer wants to check if a flight is likely to be delayed based on date, time, and airport.
2. Operational Use Case: Airline staff input multiple flight parameters to flag risky schedules.

UI/UX Considerations :

1. Simple form layout for entering flight details.
2. Clear labels and dropdowns for categorical inputs(e.g., Origin, Destination).
3. Prediction result shown as a card: "On Time" or "Delayed".
4. Responsive layout for both desktop and mobile viewing.
5. Real-time alert for missing or invalid inputs.

Technologies used :

1. Frontend: HTML, JavaScript
2. Backend: Python, Flask
3. Model: Scikit-learn trained classifier (e.g., Random Forest)
4. Deployment: Render(backend), GitHub Pages(frontend)
5. Storage: .pkl file for trained ML model

Flight Delays Prediction Using Machine Learning : Project Planning

Development Phases :

1. Requirement Gathering:
Identified necessary inputs(e.g.,origin,destination,date,times) based on available flight datasets and user needs.
2. Data Preprocessing and Model Training:
Cleaned and transformed data using techniques like one-hot encoding.Trained a classification model (flight.pkl) using scikit-learn.
3. Frontend UI Design:
Developed a responsive form using HTML for users to input flight details easily.
4. Backend Development:
Implemented a Flask server to handle form submissions,process input data,load the ML model and return predictions.
5. Testing & Validation:
Ran multiple test cases to validate accuracy,check model response time and ensure proper error handling.
6. Deployment:
Hosted the backend on Render.The frontend is served at GitHub Pages.

Project Timelines :

Phase	Duration	Description
Planning and Ideation	1-2 days	Defining scope and

		dataset
Data cleaning and modeling	2-3 days	Train and validate ML model
Frontend development	1-2 days	Build input UI using HTML
Backend Integration	1-2 days	Set up Flask and load the model
Testing	1-2 days	Test inputs and to calculate accuracy
Deployment	1 day	Hosting app on render

Team Member roles :

1. ML Engineer (V.V.Ramanji) : Handles loading a dataset, preprocessing, model training and optimization.
2. Backend Developer (T. Hema Sri) : Builds and maintains flask integration and make sure that input form is taking responses from html page.
3. Frontend Developer(V. Yasaswi Venkat) : Designs UI for form inputs and displays prediction results.
4. Project Lead (T. Sandeep) : Coordinates all the phases, help team members and manages deployment.