FLIGHT FARE PREDICTION

LOW LEVEL DESIGN

2024

MACHINE LEARNING PROJECT

MANVENDRAS2608@GMAIL.COM

Document Version Control

DATE ISSUED	VERSION	DESCRIPTION	AUTHOR
20/02/2024	0.0.1	PROJECT STRUCTURE	MANAV SINGH
22/02/2024	0.0.2	DATA PREPROCESSING AND MODEL BUILDING	MANAV SINGH
23/02/2024	1.0.0	FINISHED MODEL TRAINING AND DEPLOYMENT	MANAV SINGH

CONTENTS

Abstract

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. The flight ticket prices increase or decrease every now and then depending on several factors like timing of the flights, destination, and duration of flights on various occasions such as vacations or festive season. Therefore, having some basic idea of the flight fares before planning the trip will surely help many people save money and time. This System is designed to predict flight fares using machine learning algorithms based on various input parameters such as airline, source, destination, departure time, arrival time, duration, and total stops. This system integrates a web application for user interaction, allowing users to input their travel details and receive fare predictions.

1. Introduction

1.1 WHY THIS LOW-LEVEL DESIGN DOCUMENT?

This document aims to provide a comprehensive overview of the Flight Fare Predi ction System. It will detail the system's objectives, functionalities, user interfaces, o perational constraints, and response to external inputs. Targeted at both stakehold ers and developers, this document is crafted to facilitate a clear understanding of t he project's scope and technicalities. It will serve as a foundational piece for discus sions and approval by the management team.

1.2 SCOPE

The primary goal of this project is to develop a predictive model capable of estimat ing the fare of flight tickets based on various factors. These factors include, but are not limited to, departure and arrival times, dates, destinations, the number of stop s, and the airline. The system leverages historical flight data to train its predictive model, ensuring accurate and reliable fare predictions.

- Flight Fare Prediction Systems play a crucial role in the travel and aviation industry by offering:
- Predictive Insights: Utilizing historical data to forecast future flight prices, aiding budget planning for travelers and pricing strategies for airlines.
- Data-Driven Decision Making: Providing access to analytical tools that help in making informed decisions regarding flight bookings and travel plans.
- Optimization of Travel Expenses: Helping users find the best possible fares by analyzing patterns and trends in flight pricing.
- The system encompasses a wide range of data, including:
- Flight schedules and timings
- Airline information
- Pricing history
- Seasonal and temporal factors affecting flight prices.

2. General Description

2.1 PROJECT PERSPECTIVE

The Flight Fare Prediction System is designed to predict flight fares using machine learning algorithms based on various input parameters such as airline, source, destination, departure time, arrival time, duration, and total stops. This system integrates a web application for user interaction, allowing users to input their travel details and receive fare predictions.

2.2 DATA GATHERING

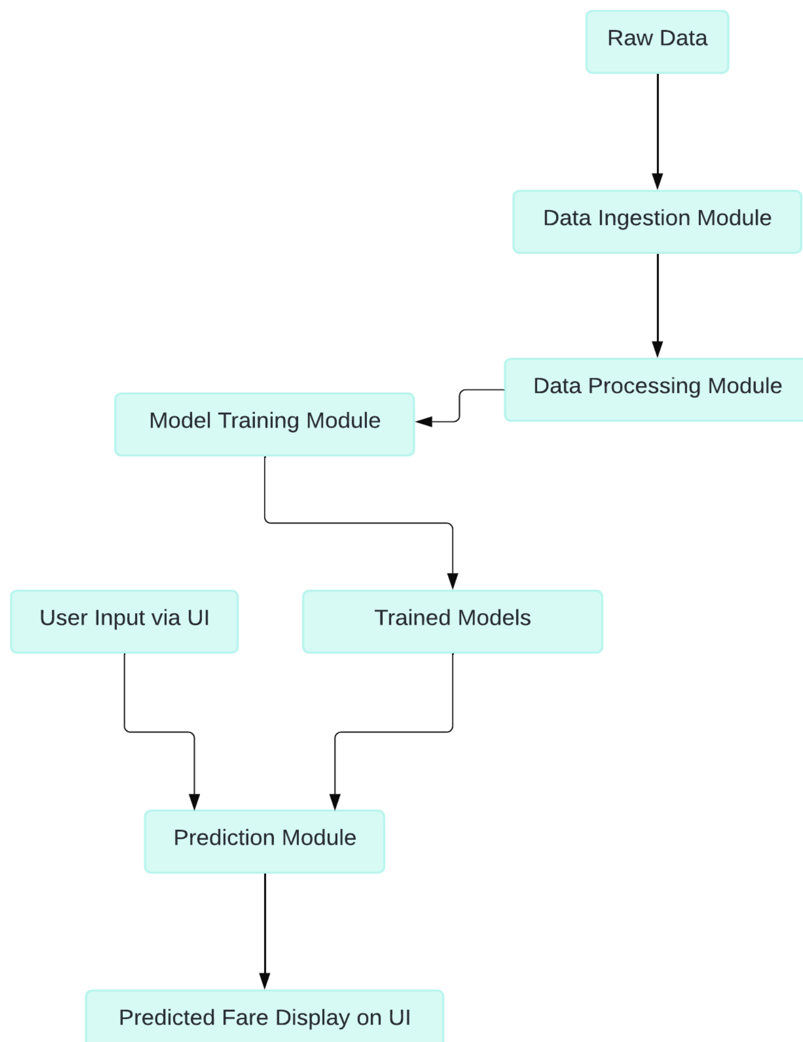
The dataset for this project was sourced from Kaggle.
It can be accessed and downloaded from the following link: Flight Fare Prediction Dataset**.**

Features Available in the Dataset are:

- **Airline:** The name of the airline.
- **Date of Journey:** The date of the flight.
- **Source:** The starting point of the journey.
- **Destination:** The endpoint of the journey.
- **Route:** The route taken by the flight.
- **Dep_Time:** The time when the flight departs.
- **Arrival Time:** The time when the flight arrives.
- **Duration:** Total duration of the flight.
- **Total_Stops:** Total stops between the source and destination.
- **Additional_Info:** Additional information about the flight.
- **Price:** The price of the flight ticket.

2.3 SYSTEM ARCHITECTURE

- **Data Ingestion Module:** Responsible for ingesting historical flight data from various sources.
- **Data Processing Module**: Cleanses and preprocesses the data, handling missing values, outliers, and feature engineering.
- **Model Training Module**: Trains predictive models using processed data.
- **Prediction Module:** Uses trained models to predict flight fares based on user input.
- **API Layer**: Exposes endpoints for data ingestion, model training, and fare prediction.
- **User Interface (UI):** Provides a web interface for users to input prediction parameters and view results.



2.4 TOOLS USED

- **Python**: Primary programming language for data preprocessing, model training, and web application backend.
- **Pandas and NumPy:** For data manipulation and numerical computations.
- **Scikit-learn**: For machine learning model training and evaluation.
- **Flask:** For developing web applications.
- **HTML/CSS:** For designing the web application's frontend.
- **GitHub:** For version controlling.
- **Visual Studio Code:** It is used as IDE.

3. Detailed Design

In the Flight Fare Prediction project, model training involves several key steps to ensure the machine learning model accurately predicts flight fares based on input features. Here is a more in-depth explanation of the model training process as applied in this project.

3.1 DATA PREPROCESSING

- **Converting date columns** to datetime format to extract useful features such as day and month.
- **Handling missing values** by dropping rows with missing data, as the dataset is large enough to afford losing a small fraction of data.
- **Encoding categorical variables** like Airline, Source, and Destination using one-hot encoding to convert them into a format that can be provided to the model.
- **Feature engineering** to create new features that might be relevant for the prediction, such as extracting the day of the week from the date of journey.

3.2 FEATURE SELECTION

The project selects features that are deemed relevant for predicting flight fares. This includes:

- **Numerical features** like Duration of the flight.
- **Categorical features** that have been one-hot encoded.
- **Newly engineered features** such as DAY and MONTH extracted from the DATE_OF_JOURNEY column.

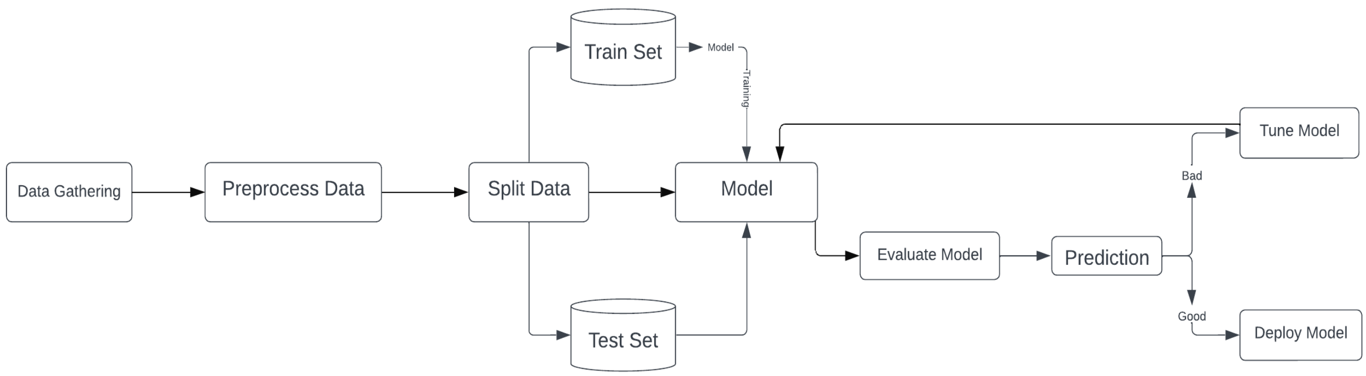
3.3 MODEL SELECTION AND HYPERPARAMETER TUNING

- The project employs **RandomizedSearchCV** for hyperparameter tuning, which is a more efficient approach than **GridSearchCV** when dealing with a large number of hyperparameters and data.
- **Multiple regression models** are evaluated, including potentially **Linear regression, Decision trees, Random forests,** and **Gradient boosting** machines, though the exact models used are not specified in the provided inputs.

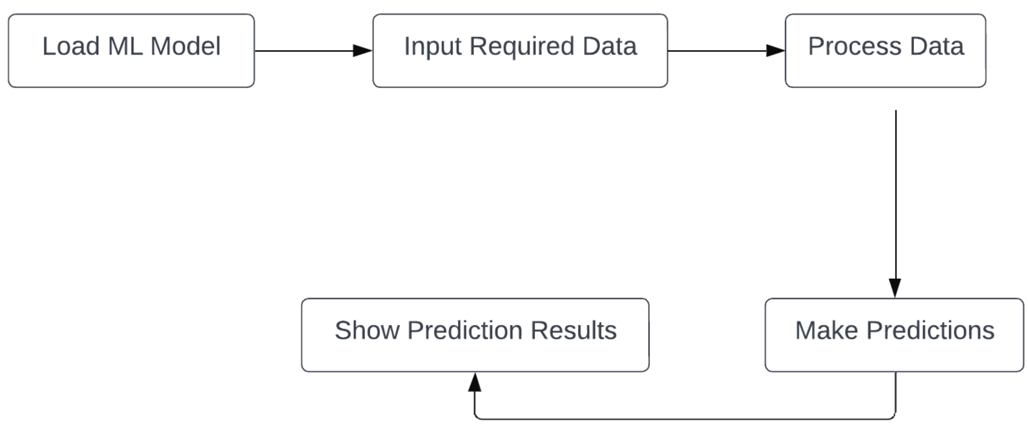
3.4 MODEL TRAINING AND EVALUATION

- The selected model is trained on the preprocessed training dataset. This involves learning the relationship between the input features and the target variable (**PRICE**).
- **Cross-validation** is used during hyperparameter tuning to ensure the model's generalizability and to prevent overfitting.
- The model's performance is evaluated using the **R2 score**, which measures the proportion of variance in the dependent variable that is predictable from the independent variables. This is a common metric for regression tasks.
- The evaluation is done on a separate test set that the model has not seen during training to assess its performance on unseen data.

Model Training and Evaluation Flow Chart



Model Deployment (Localhost)



3.5 EVENT LOG

The system should log every event so that the user will know what process is running internally.
Initial Step-By-Step Description:

- The System identifies at what step logging is required.
- The System should be able to log every system flow.
- Developers can choose logging methods. You can choose database logging/ File.
- logging as well.
- System should not hang even after using so many loggings. Logging just because
- We can easily debug issues, so logging is mandatory to do.

3.6 ERROR HANDLING

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.7 REUSABILITY

The code written and the components are reusable and could be reused without any problems.

3.8 FUTURE ENHANCEMENTS

- Incorporating additional features such as weather conditions and holidays for improved prediction accuracy.
- Adding user authentication and personalized recommendations.
- Enhancing the web application's user interface for a better user experience.

Conclusion

The Flight Fare Prediction System leverages machine learning and web development technologies to provide users with accurate fare predictions, enhancing the flight booking experience.