FLIGHT FARE PRIDICTION

HIGH LEVEL DESIGN

2024

MACHINE LEARNING PROJECT

MANVENDRAS2608@GMAIL.COM

https://github.com/singh-manavv/Flight-Fare-Prediction

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 HIGH-LEVEL DESIGN DOCUMENT?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a prominent level.

The HLD will:

- Present all the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the hardware and software interfaces.
- Describe the performance requirements.
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 SCOPE

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

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 PROBLEM STATEMENT

To create a Machine Learning model that will predict the fares of the flights based on several factors available in the provided dataset. The factors 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 APPROACH

The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that are best fit for the above case.

2.4 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.

2.5 SYSTEM OVERVIEW

The Flight Fare Prediction project works by following these steps:

- **Data Preprocessing**: Cleansing and preparing the data for training, including handling missing values, encoding categorical variables, and normalizing the data.
- **Feature Selection**: Selecting the most relevant features that influence flight prices.
- **Model Training**: Using processed data to train a machine learning model. Various algorithms like Random Forest, Gradient Boosting, and Linear Regression can be explored to find the best performer.
- **Model Evaluation**: Evaluating the model's performance using metrics like MAE (Mean Absolute Error) and R^2 score.
- **Prediction**: Using the trained model to predict flight fares based on user input.

2.6 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.


NumPy


scikit
learn


HTML


CSS


git

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.

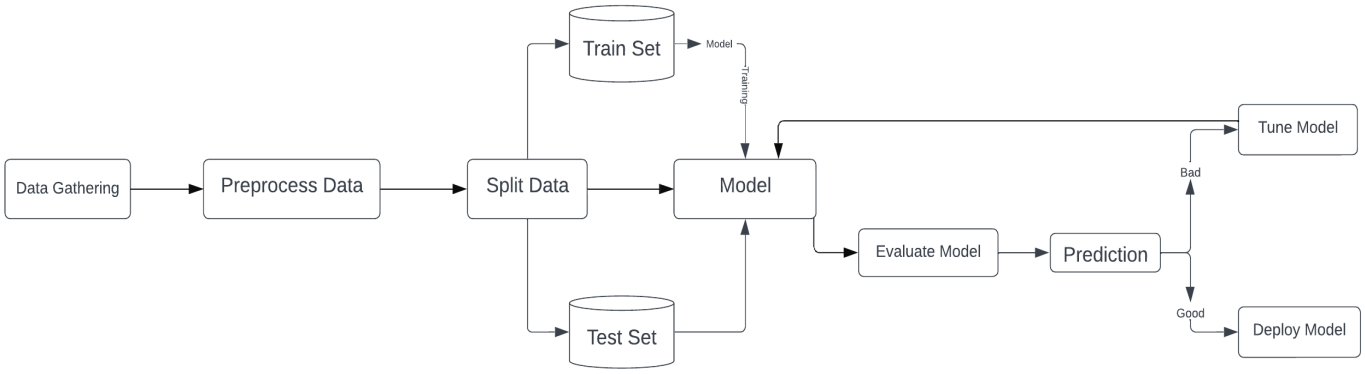
3.3 MODEL SELECTION AND HYPERPARAMETER TUNING

- 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.

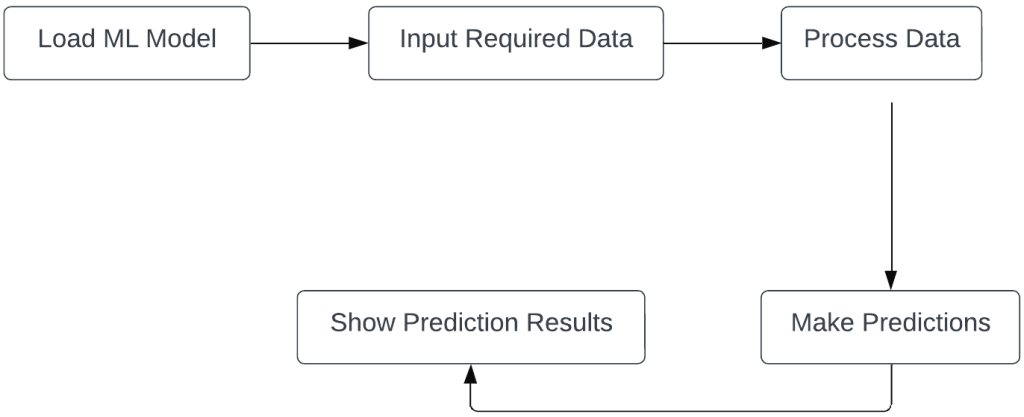
3.4 MODEL TRAINING AND EVALUATION

- 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.