Low Level Design (LLD)

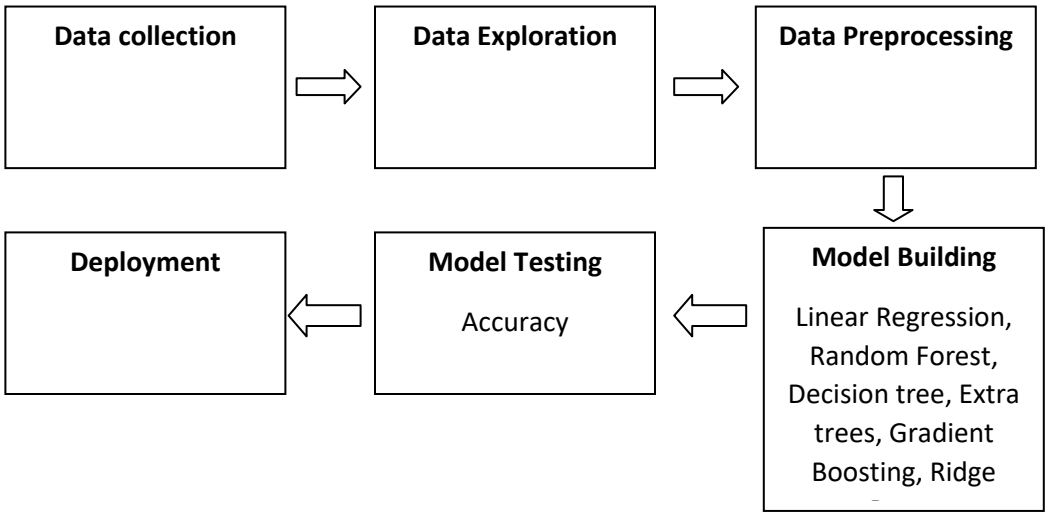
Flight Fare Prediction

Table of contents

1. Introduction

The Low-Level-Design (LLD) gives an understanding of the internal logic design of the code for the flight fare prediction system. The LLD highlights different aspects of the architecture, the data involved and the various steps undertaken to reach the final goal of the project.

2. Architecture



3. Architecture Description

3.1 Data Description

The dataset used for the project is the Flight Fare Prediction dataset by MachineHack which is in a .csv format consisting of 10683 rows of data relating to different airlines and 11 variables like name of the airline, source and destination, date of departure and arrival, price of the ticket, duration of flight and number of stops

3.2 Import Data

Data is stored and imported to Python in CSV format which is then used for data pre-processing and model training and testing

3.3 Data Pre-processing

The dataset was fairly clean and had very negligible amount of null values where the concerned rows were removed as removing them would not affect the quality of the output. The dataset had some duplicated rows which were removed as keeping them would skew the data. The destination column contained instances of both “Delhi” and “New Delhi” which was merged as “Delhi”. Categorical columns like the name of the airline, source, destination and number of stops had to be encoded to make them suitable for the regression model. The columns relating to dates were converted to date-time format to extract the day and month of travel. Columns like Additional info, route , duration of flight, date of arrival were removed as when we look from the standpoint of actually booking a flight, the variables actually involved in displaying the fare of the flight are date and month of travel, name of airline, source and destination and number of stops and departure time. Before building the models, the categorical columns were encoded using Label Encoder so that each airline, source and destination were given their separate key.

3.4 Splitting the data

The dataset was split into train-test sets with the independent variables being the different features and the dependent variable being the price of the flight

3.5 Model Building

After cleaning and splitting the data, the machine learning models were built to understand which model best predicts the price of the flights based on R2 score. The algorithms built for the task were Linear Regression, Random Forest, Decision tree, Extra trees regressor, Gradient boosting and Ridge regression which were then trained on the training dataset

3.6 Model Testing

The models were then tested on the testing dataset and the model that predicted the dependent variable with the highest R2 score was treated as the best model.

3.7 Deployment

The model will be deployed as an API using FastAPI where the user can input the relevant values to find the fare price of their preferred flight.