Low Level Design

Flight Fare Prediction

Written By	ANKIT, MUDIT ARORA, MS.S.S.CHAITANYA
Document Version	0.3
Last Revised Date	23 – APRIL -2023

Document Control

Change Record:

Version	Date	Author	Comments
0.1	19 – APRIL -2023	M.S.S.S.CHAITANYA	Introduction & Architecture defined
0.2	20 – APRIL- 2023	ANKIT	Architecture & Architecture Description appended and updated
0.3	21 – APRIL -2023	MUDIT ARORA	Unit Test Cases defined and appended

Reviews:

Version	Date	Reviewer	Comments
0.2	21 – APRIL - 2023		Document Content , Version Control and Unit Test Cases to be added

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

Abstract

The Aviation sector faced significant challenges due to the recent changes in the international market. These changes affected both the Business and Customer perspectives of the industry. One of the main reasons for this was the different regulations imposed by governments around the world on their Airline companies. As a result, the flight ticket prices varied widely across different locations. There were also two modes of booking a flight ticket: online and offline. Each mode had its own factors that influenced the price, such as the server load and the number of booking requests. In this machine learning implementation, we will explore various factors that affect the flight ticket price and predict the optimal price for the ticket.

1. Introduction

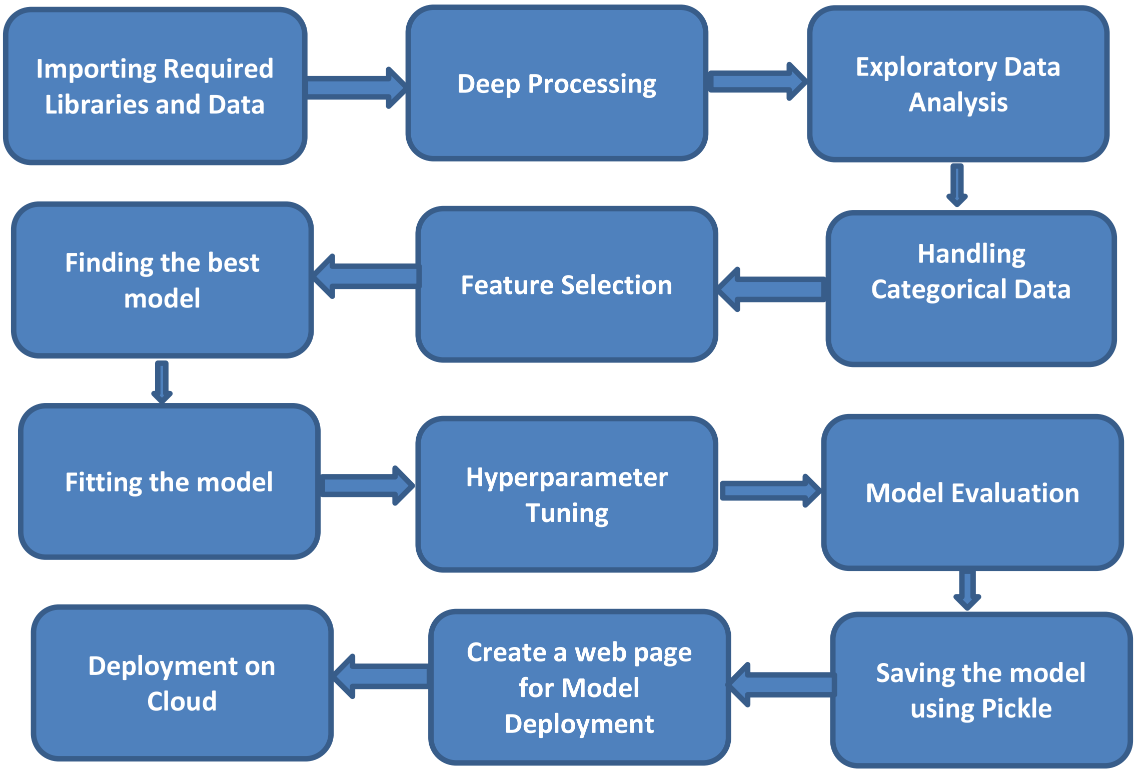
1.1 What is Low-Level design document?

The main goal of the LLD document is to give the internal logic design of actual code implementation and supply the outline of the machine learning model and its implementation. Additionally, it provides the description of how our project will be designed end-to-end.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture


Importing Required Libraries and Data
Deep Processing
Exploratory Data Analysis
Handling Categorical Data
Feature Selection
Finding the best model
Fitting the model
Hyperparameter Tuning
Model Evaluation
Saving the model using Pickle
Create a web page for Model Deployment
Deployment on Cloud

2.1 Architecture Design

This project is designed to make an interface for the User to predict its approximate flight ticket price.

2.2 Data Description

The data for these project is collected from the Kaggle Dataset, the URL for the data set is kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh

Flight Fare Prediction is a 10K+ data set publicly available on the Kaggle. The information in the data set is present in two separate excel files named as train.xlsx and test.xlsx. The data set contains 10683 rows which shows the information such as Date of Journey, Source, Destination, Arrival Time, Departure Time, Total stops, Airlines, Additional Info, and Price. A glance of the Data set is :

2.3 Importing data into Database

Created associate API for the transfer of the info into the Cassandra info, steps performed are :

- Connection is created with the info.
- Created info with the name Flight Info.
- cqlsh command is written for making the info table with needed parameters.
- And finally, a cqlsh command is written for uploading the Knowledge Set into the data table by bulk insertion.

2.4 Exporting Data from Database

In the above-created API, the download URL is also being created, which downloads the data into a CSV file format.

2.5 Data Pre processing

- Checked for info on the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Converted all the desired columns into Date time format.
- Performed One – Hot encoding on the desired columns.
- Checking the distribution of the columns to interpret their importance.
- Now, the info is prepared to train a Machine Learning Model.

2.6 Model Creation

The Pre processed info is now envisioned and drawn insights help us to select the feature that improves the accuracy of the model. The info is randomly used for modeling with different machine learning algorithms to create a model to predict the Flight ticket price. After performing on different algorithms, we use Random Forest Regression to create a model and then also perform Hyper parameter Tuning to improve the accuracy of the model.

2.7 Data from User

The data from the user is retrieved from the created HTML web page.

2.8 Data Validation

The data provided by the user is then processed by the app.py file and validated. The validated data is then sent to the prepared model for prediction.

2.9 Rendering the Results

The data sent for the prediction is then rendered to the web page.

3. Deployment

The tested model is then deployed to Railway. So, users can access the project from any internet device.

4. Unit Test Cases

Test Case Description	Pre - Requisites	Expected Results
Verify whether the Web page is accessible to the User or not.	Web page URL should be defined.	Web page should be accessible to the User.
Verify whether the Web page is completely loads for the User or not.	Web page URL is accessible. Web page is deployed.	The Web page should be completely loads for the User when it is accessed.
Verify whether the user is able to enter data in input fields or not.	Web page URL is accessible. Web page is deployed. Web page input fields are editable.	The User is able to enter data in input fields.
Verify whether the user is able to submit details or not.	Web page URL is accessible. Web page is deployed. Web page input fields are editable.	The User is able to submit details to process.
Verify whether the user gets recommended results on submitting the details or not.	Web page URL is accessible. Web page is deployed. Web page input fields are editable.	The User gets recommended results on submitting the details.