Low Level Design

Flight Fair Prediction

Written By	Biswajit Jena
Document Version	0.3
Last Revised Date	26 – july -2024

Document Control

Change Record:

Version	Date	Author	Comments
0.1	20 – July - 2024	Biswajit Jena	Introduction & Architecture defined
0.2	23 – July - 2024	Biswajit Jena	Architecture & Architecture Description appended and updated
0.3	26 – July - 2024	Biswajit Jena	Unit Test Cases defined and appended

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

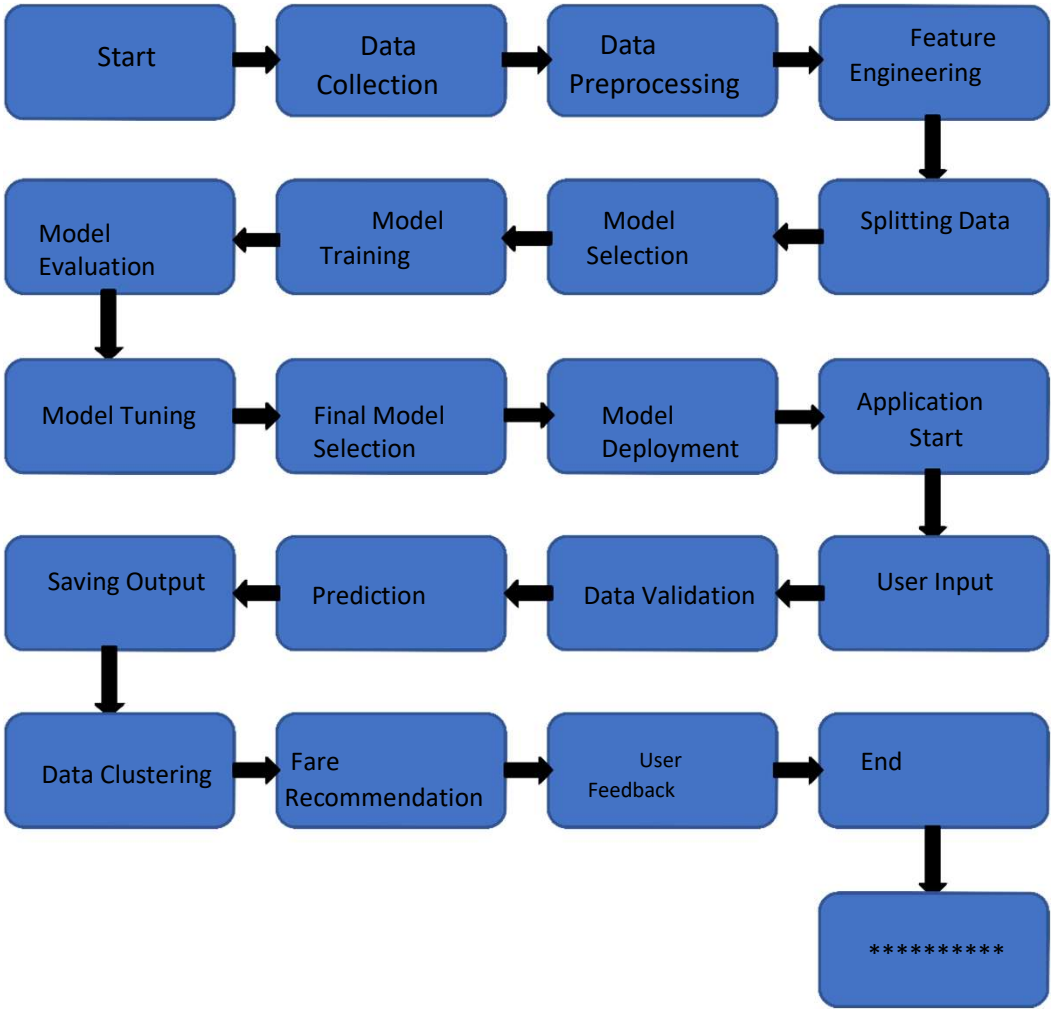
1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

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



3. Architecture Description

Data Collection
This step involves gathering the necessary data for building the model. This data can be sourced from various places such as web scraping, APIs, or available datasets. For a flight fare prediction project, this might include historical flight prices, flight schedules, airline information, and more.

Data Preprocessing
In this step, the collected data is cleaned and prepared for analysis. This involves handling missing values, removing duplicates, correcting errors, and normalizing or scaling numerical features. For example, if certain flight records have missing fare values, these need to be addressed to avoid inaccuracies in the model.

Feature Engineering
Feature engineering involves creating new features from existing data to improve the model's predictive power. This can include deriving new variables, such as the duration of the flight, the time of the day the flight departs, or creating binary variables for categorical data like airline names.

Splitting Data
The preprocessed data is split into training and testing sets. The training set is used to build the model, while the testing set is used to evaluate its performance. A common split is 80% for training and 20% for testing, ensuring the model generalizes well to unseen data.

Model Selection
Various machine learning models are considered for predicting flight fares, such as Linear Regression, Decision Trees, Random Forests, Gradient Boosting, or even more advanced methods like Neural Networks. The goal is to identify which model performs best for the given data and problem.

Model Training
The selected model(s) are trained on the training data. This involves feeding the features into the model and adjusting its parameters to minimize prediction errors. For example, training a Random Forest model would involve fitting multiple decision trees to the data.

Model Evaluation
The trained model is evaluated using the testing set to measure its performance. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared. This step helps in understanding how well the model can predict unseen data.

Model Tuning
Hyperparameters of the model are fine-tuned to optimize performance. Techniques like Grid Search, Random Search, or Bayesian Optimization are used to find the best combination of hyperparameters. For instance, adjusting the number of trees in a Random Forest or the learning rate in a Gradient Boosting model.

Final Model Selection
After tuning, the best-performing model is selected as the final model. This model will be used for making predictions on new data. The selection is based on the model's evaluation metrics and its ability to generalize well to new data.

Model Deployment
The final model is deployed to a production environment where it can be used by end-users. This involves setting up the necessary infrastructure, such as servers and databases, to host the model and handle user requests.

Application Start
The application that uses the model is started, making it accessible to users. This might involve launching a web application, a mobile app, or any other interface through which users can interact with the model.

User Input
Users provide input data through the application. For flight fare prediction, this might include details such as departure and arrival locations, travel dates, preferred airlines, and so on.

Data Validation
The input data from users is validated to ensure it is complete, correctly formatted, and within expected ranges. For example, checking that the travel dates are valid and the locations exist.

Prediction
The validated data is fed into the deployed model to generate predictions. The model processes the input and outputs the predicted flight fare.

Saving Output
The predicted fare is saved to a database or any other storage system, ensuring the results are available for future reference or analysis.

Fare Recommendation
Based on the predicted fare, the application might provide recommendations to the user, such as suggesting alternative dates or airlines for better prices.

User Feedback
Users provide feedback on the prediction and recommendation. This feedback can be used to further improve the model. For example, if users consistently find the fare predictions inaccurate, the model might need retraining with additional or more relevant data.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	- Application URL should be defined	- Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1.Application URL is accessible 2.Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1.Application is accessible 2.User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The recommended results should be in accordance to the selections user made
Verify whether user has options to filter the recommended results as well	1.Application is accessible 2.User is signed up	User should have options to filter the recommended results as well

	to the application 3. User is logged in to the application	
Verify whether KPIs modify as per the user inputs for the Flight	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	KPIs should modify as per the user inputs for the Flight
Verify whether the KPIs indicate details of the suggested Flight Price	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The KPIs should indicate details of the suggested Flight Price