Low Level Design Document (LLD)

Flight Fare Prediction - System

Document Control

Change Record:

Version	Date	Author	Comments
0.1	04 – Jan - 2022	Abhishek Saha	Introduction & Architecture defined
0.2	06 – Jan - 2022	Abhishek Saha	Architecture & Architecture Description append
0.3	08 – Jan - 2022	Abhishek Saha	Unit Test Cases defined and append

Contents

1. Introduction

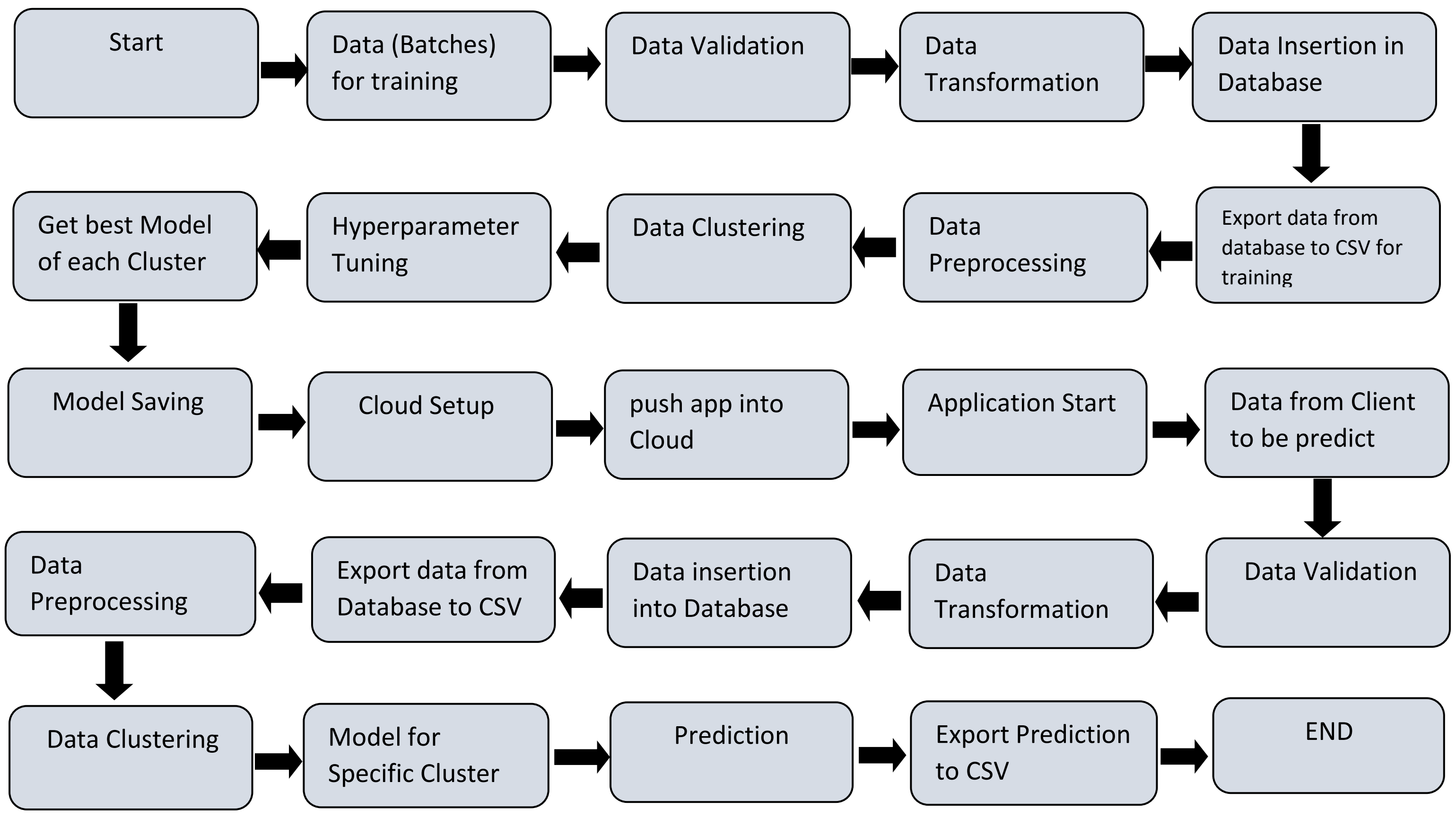
1.1. What is Low-Level design document?

The goal of LLD or low-level design document (LLDD) is to give the internal logical design of the actual program code for Flight Fare Prediction System. LLD describes the class diagram 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.

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



3. Architecture Description

3.1 Data Description

Receive 5 CSV file as a dataset for training. Each Dataset contain above 10,000 number of rows and 11 number of columns. Each column contains several information like Flight name, Departure time, Arrival time, Date of Journey, Source, Destination, Duration, Route, Number of Stoppage, Additional Details and Fare.

For Prediction receive 5 CSV files for prediction. Each Dataset contain above 2000 number of rows and 10 number of columns. All Columns like training dataset have except Label feature (Price).

3.2 Dataset Information:

Airline: Different Flight provided service.
Date-of-Journey: Passenger Journey Date
Source: Place where Flight will start journey.
Destination: Passenger Destination.
Route: Route follow by the Flight to travel from source to destination.
Dep_time: when Flight will start journey.
Arrival Time: Time when Flight reached the destination.
Duration: Time taken to travel from source to destination.
Total Stop: Number of stops in the journey.
Additional info: It provide the information regarding the Flight organization provide extra service.

3.3 Data from User for Training:

User Provide Master Data Management of the dataset as well as provide batch of files.

3.4 Data Transformation

In the Transformation Process, replace all NaN value by NULL and typecast all data into String datatype.

3.5 Data Insertion into Database

a. Database Creation and connection name is decided based on training and prediction. If the database is already created, open database connection and return connection object otherwise create a new database on same and return the connection object.
b. Table creation in the database.
c. Insertion of files in the table

3.6 Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

3.7 Data Pre-processing

Data Pre-processing steps Initially load data in CSV file. Then perform Data pre-processing on CSV files, technique used to perform Data Pre-processing are Removing duplicate rows, Separate Label feature, Null value handling, formatting date and time feature, Label Encoding also drop some columns etc.

3.8 Data Clustering

K-Means algorithm is used to create clusters for the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot and for the dynamic selection of the number of clusters, we are using KneeLocator function. The idea behind clustering is to implement different algorithms based on each clusters data. The K-Means Model is trained over preprocessed data and the K-Means Model is saved for further use in prediction dataset to get separate cluster based on characteristic of data.

3.9 Model Building

After create cluster, find the best Model between Random-Forest-Regression or XGBoost for each cluster. For each cluster, find the best Model by RandomizedSearchCV Hyper-parameter tuning. Then calculate the R-Square and Adjusted_R_Square score for each Model and select the Model with the best score and save it for prediction. This way, the Models will be selected for each cluster. All the Models for every cluster will be saved for use Flight Fare Prediction on predict dataset.

3.10 Data from User for prediction

Here we will collect CSV file as well as user can provide their local system file path to Predict the final outcome.

3.11 Data Validation

Process data validation steps on Prediction dataset.

3.12 User Data Inserting into Database

Collecting the data from the user and storing it into the database. The database can be either SQLite or Mongo DB.

3.13 Data Clustering

Cluster Model created during training. Now load the saved cluster Model, and cluster Model process on the user data to predict cluster.

3.14 Model Call for Specific Cluster

Based on the cluster number, the respective Model will be loaded and will be used to find the predicted Fare for that cluster data.

3.15 Flight Fare Prediction result and Saving Output

Process the predict data by the Model and generate the output, this output will be saved in CSV format and it will be store as per the User Specified location.

3.16 Deployment

We are deploying the application on Heroku Cloud Platform.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. 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 user gets Submit button to submit the inputs	1. Application is accessible 2. 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 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 logged in to the application	The recommended results should be in accordance to the selections user made