Low Level Design (LLD)

Flight Fare Prediction

HARSHA MS

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | **Version** | **Description** | **Author** |
| 24/10/2022 | 1.0 | Introduction & Architecture defined | Harsha MS |
| 27/10/2022 | 1.1 | Architectural Design | Harsha MS |
| 30/10/2022 | 1.2 | Unit Test Cases | Harsha MS |

**Contents**

Document Version control----------------------------------------------------------------2

1 Introduction-------------------------------------------------------------------------------4

1.1 What is Low-Level Design Document-------------------------------------4

1.2 Scope-----------------------------------------------------------------------------4

2 Architecture------------------------------------------------------------------------------5

3 Architecture Description--------------------------------------------------------------- 6

3.1 Data Collection ---------------------------------------------------------------6

3.2 Data Description--------------------------------------------------------------6

3.3 Export Data from Database to CSV for Training-------------------------6

3.4 Data Pre-processing ----------------------------------------------------------6

3.5 Model Creation----------------------------------------------------------------6

3.6 Data from the User-------------------------------------------------------------6

3.7 Data Validation -----------------------------------------------------------------7

3.8 Rendering the results-----------------------------------------------------------7

4 Deployment--------------------------------------------------------------------------------8

4.1 Unit Test Cases-----------------------------------------------------------------8

**Abstract**

The recent changes in the international market had a large impact on the Aviation sector because of the several reasons. These impact the two class folks, the first is the Business perspective and the second is Customer perspective. The major reason of such impact is the governments around the world amended totally different rules to their various Airline firms. Taking of these factors in thought the value of the flight tickets has vary from one place to another. Booking a flight ticket its price tag has split into two, one is online bookings and other is offline bookings. Each of these have their various criteria for the value of the price, one such example is that the server load and therefore the range of booking requests. During this machine learning implementation, we are going to see numerous factors that impact the price of the flight ticket and predict the acceptable price of the ticket.

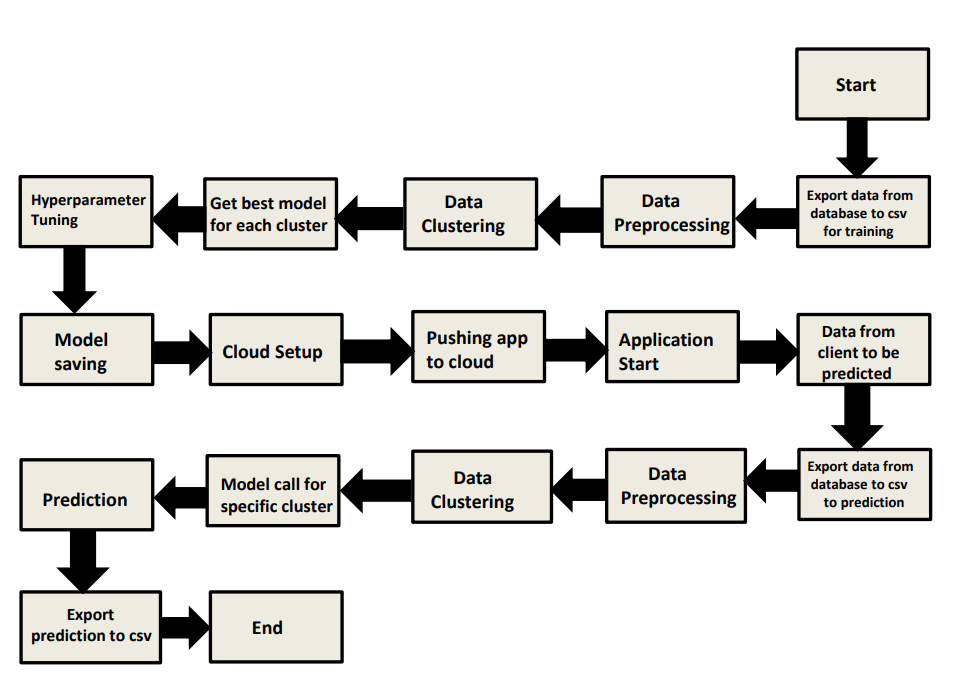
1. **INTRODUCTION**
   1. **What is Low-Level Design document?**

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. **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.

1. **ARCHITECTURE**

****

1. **ARCHITECTURE DESIGN**

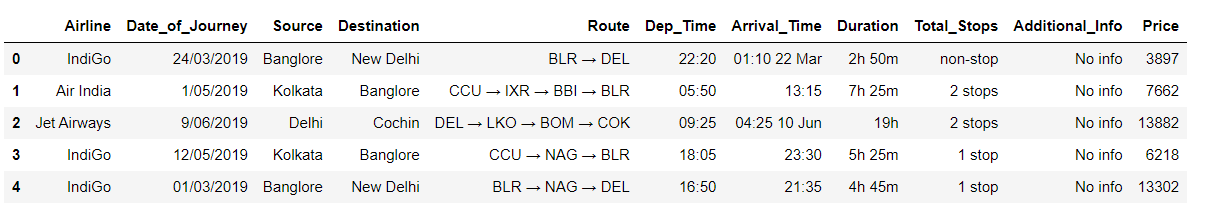
This project is designed to make an interface for the User to predict its approximate flight ticket price.

* 1. **Data Collection**

The data for this project is collected from the Kaggle Dataset, the URL for the dataset is <https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh> <https://astra.dev/ineuron>

* 1. **Data Description**

Flight Fare Prediction is a 10K+ dataset publicly available on the Kaggle. The information in the dataset is present in two separate excel files named as train.xlsx and test.xlsx. The dataset 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 Dataset is:



* 1. **Export Data from Database to CSV for Training**

All those batch file which after passing through data validation step get stored in Database. And here we will be exporting all those CSV files in a single CSV file for training purpose. For this we need to write a separate module.

* 1. **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.

* 1. **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 modelling 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 Hyperparameter Tuning to improve the accuracy of the model.

* 1. **Data from the User**

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

* 1. **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.

* 1. **Rendering the Results**

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

1. **Deployment**

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

* 1. **Unit Test Cases**

|  |  |  |
| --- | --- | --- |
| Test Case Description | Pre-Requisite | Expected Result |
| Verify whether the  Webpage is accessible to the User or not. | Webpage URL should be defined. | Webpage should be accessible to the User |
| Verify whether the  webpage is completely loads for the User or not. | Webpage URL is accessible.  Webpage is deployed. | The Webpage 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. | Webpage URL is accessible  Webpage is deployed.  Webpage 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 | Webpage URL is accessible.  Webpage is deployed.  Webpage 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. | Webpage URL is accessible.  Webpage is deployed.  Webpage input fields are editable. | The User gets recommended results on submitting the details. |