# High Level Design (HLD)

## FLIGHTFARE PREDICTION

Document Version: 0.3

Last Revised Date: 14-06-2023

Rajesh Singh Gurjar

Document Version Control:

| Version | Date | Author | Description |
|---|---|---|---|
| 0.1 | 10-06-2023 | Rajesh | Introduction, Problem-Statement |
| 0.2 | 11-06-2023 | Rajesh | Design Flow |
| 0.3 | 14-06-2023 | Rajesh | Performance Evaluation |

# Contents

# 1. Introduction

## 1.1. What is High-Level design document?

The main purpose of this HLD document is to feature the required overview of the project and provide the outline of the different stages of machine learning end-to-end project along with its deployment.

 The HLD describe:
• The design aspects and define them in detailed.
• The user interface being implemented.
• The hardware and software interfaces.
• The performance requirements.
• Include design features and architectural design of the project.
• List and describe the non - functional attributes like:
       1. Security
       2. Reliability
       3. Maintainability
       4. Portability
       5. Reusability
       6. Resource
       7. Utilization

## 1.2. Scope

The HLD documentation presents the structure of the project, such as architectural design, application flow and database design. The HLD uses non-technical terms to technical terms that can be understood to the administrator of the system.

**Definitions**

| FFP | Flight Fare Prediction |
|---|---|
| MongoDB | MongoDB is used to store and retrieve dataset |
| VS Code | VS Code stands for Visual Studio code and it is an IDE for the development of the python based project |
| AWS | AWS is Amazon Web Services that enables developers to build, run, and operate applications entirely in the cloud. |

# 2. Description

## 2.1. Problem Perspective

The flight fare prediction may be a machine learning model that helps America to predict the price of the flight price tag and helps the users to understand the price of their journey.

## 2.2.   Problem Statement

The flight fare prediction problem involves using historical flight data to predict the future cost of a flight. This data can include factors such as the date of the flight, the departure and arrival airports, the airline, and the class of service. The goal of the prediction is to provide accurate information on flight costs to consumers in order to assist with flight planning and pricing decisions.

## 2.3.   Purposed Solution

We are using the machine learning regression model to predict the price of flights for the given set of cities. Implementing the ML model using modular approach and saving the latest robust model with for the current dataset. The proposed model will predict the prices using batch file.

## 2.4. Solution Improvements

We have used various regression algorithms with their hyperparameter to find out the robust model as per the given dataset moreover in the training pipeline the code will evaluate the performance of the current model with the previously saved model, if available.

## 2.5. Technical Requirements

As technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have a device that has the access to the web and the fundamental understanding of providing the input. For the backend, we need a AWS to run all the required packages to process the input and predict the desired output.

## 2.6. Data Requirements

The data requirements totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.xlsx). Because the main theme of the project is to induce the expertise of real time issues, we have a tendency to transform the information into the MongoDB database and the csv format.

## 2.7. Tool Used

• Python 3.8 is used while creating the environment and libraries like NumPy, Pandas, Scikit-learn, Pymongo, Exceptions and logger are used for developing the model.
• VS code is used to development the modular model.
• MongoDB is used to store and retrieve the data.
• GitHub is employed for continuous deployment.
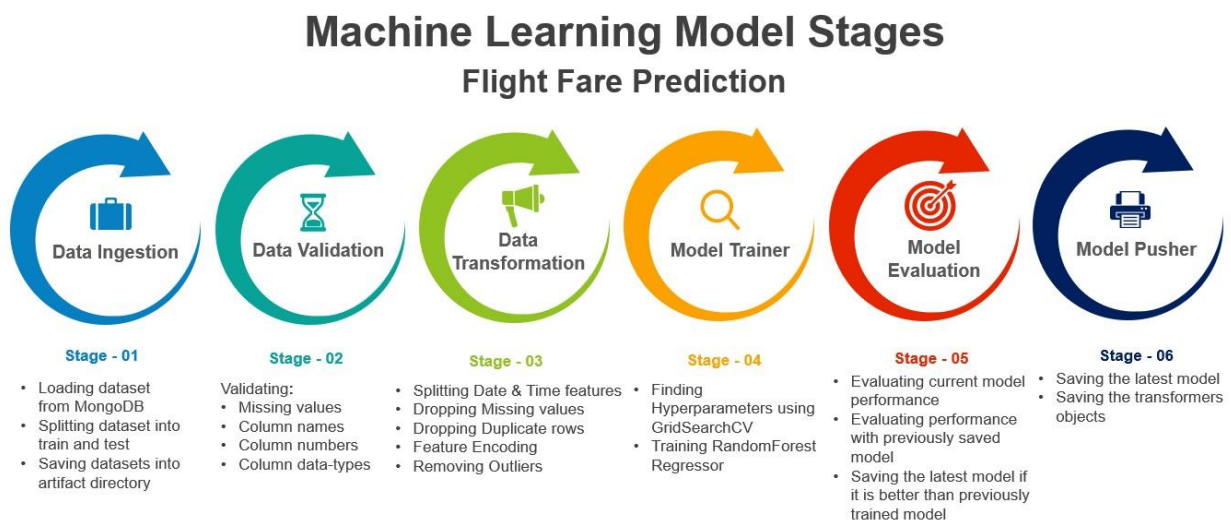• AWS is used for the deployment.

## 2.8. Constraints

The flight fare prediction answer should be user friendly, as automatic as attainable and also the user should not be needed to understand any of the operating.

### 2.9. Assumptions

The objective of the project is to implement the regression model to predict the price of the new dataset. It assumed that all aspects of this project have the ability to work together in the way it is supposed to be.

# 3. Design Flow

### 3.1. Modelling and Deployment Process



## Machine Learning Model Stages
### Flight Fare Prediction

| Data Ingestion | Data Validation | Data Transformation | Model Trainer | Model Evaluation | Model Pusher |
|---|---|---|---|---|---|
| **Stage - 01** | **Stage - 02** | **Stage - 03** | **Stage - 04** | **Stage - 05** | **Stage - 06** |
| • Loading dataset from MongoDB<br>• Splitting dataset into train and test<br>• Saving datasets into artifact directory | Validating:<br>• Missing values<br>• Column names<br>• Column numbers<br>• Column data-types | • Splitting Date & Time features<br>• Dropping Missing values<br>• Dropping Duplicate rows<br>• Feature Encoding<br>• Removing Outliers | • Finding Hyperparameters using GridSearchCV<br>• Training RandomForest Regressor | • Evaluating current model performance<br>• Evaluating performance with previously saved model<br>• Saving the latest model if it is better than previously trained model | • Saving the latest model<br>• Saving the transformers objects |

### 3.2. Logging

All events are logged into the log file as per the execution flow is created and it also captures the error or exceptions with reason and timestamp. This helps the developer to debug the system bugs and rectify the error.

### 3.3. Exception Handling

In logging, each time an error or an exception occurs, the event is logged into the log file with reason and timestamp

# 4.  Performance Evaluation

### 4.1.  Reusability

Elements of the code written is accustomed different applications and therefore the rest is changed and be reused.

### 4.2.  Application Compatibility

The various parts for this project are exploitation python as associate interface between them. every element can have its own tasks to perform, and it's the work of the python to make sure correct transfer of data.

### 4.3.  Resource Utilization

Once any task is performed, it'll doubtless use all the process power offered till that perform is finished.

### 4.4.  Deployment

The model is being deployed on AWS EC2 machine.

# 5.  Conclusion

The flight fare prediction system will estimate the flight fare based on the trained dataset in the algorithm. As a result, the user will be able to determine the approximate cost for their journey.