

High Level Design (HLD)

FLIGHT FARE PREDICITONS

Revision Number: 2.0

Last date of revision: 23/07/2023

Document Version Control

Date Issued	Version	Description	Author
17/07/2023	1	Initial HLD – V1.0	C N Divya Lakshmi
23/07/2023	2	Updated – V1.1	C N Divya Lakshmi

Contents

Document Version Control.....	2
Abstract.....	4
1 Introduction	5
1.1 Why this High-Level Design Document?.	5
1.2 Scope.	5
1.3 Definitions	5
2 General Description.....	6
2.1 Product Perspective.....	6
2.2 Problem statement.....	6
2.3 PROPOSED SOLUTION	6
2.4 Data Requirements	6
2.5 Accessibility and Privacy of Data.....	6
2.6 Information about Our Dataset	6
2.7 Data creditability and Integrity	6
2.8 Tools used.....	6
3 Design Details.....	8
3.1 Process Flow.	8
3.1.1 Model Training and Evaluation.....	8
3.2 Event log.....	8
3.3 Error Handling.....	9
3.4 Performance.....	9
3.5 Reusability.....	9
3.6 Application Compatibility	9
3.7 Resource Utilization	9
3.8 Deployment.	9
4 Dashboards.....	10
5 Conclusion	11

Abstract

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. The flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, and duration of flights various occasions such as vacations or festive season. Therefore, having some basic idea of the flight fares before planning the trip will surely help many people save money and time. This work shows how the prices of the flight varies based on various factors.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
 - Security
 - Reliability
 - Maintainability
 - Portability
 - Reusability
 - Application compatibility
 - Resource utilization
 - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

<i>Term</i>	<i>Description</i>
<i>Database</i>	Collection of all the information monitored by this system
<i>IDE</i>	Integrated Development Environment
<i>AWS</i>	Amazon Web Services

2 General Description

2.1 Product Perspective

The Flight Fare Prediction system is a Machine learning-based model which will help us to predict the fares of the flight in various circumstances.

2.2 Problem statement

The main task in this project is to develop a predictive model that can accurately estimate the flight fares based on the given features. This will help travelers plan their trips more effectively and make informed decisions about flight bookings.

2.3 Proposed Solution

This project aims to predict flight fares based on various features such as departure date and time, arrival date and time, total stops, airline, source, and destination. The prediction model is built using a Random Forest Regressor algorithm.

2.4 Data Requirements

The dataset used for this project is obtained from Kaggle and contains information about various flight routes, airlines, and their corresponding fares. The dataset can be accessed from the following link: [Flight Fare Prediction Dataset](#)

2.5 Accessibility and Privacy of Data

The dataset used is publicly accessible from Kaggle, and it does not contain any sensitive or private information about individuals.

2.6 Information about Our Dataset

The dataset consists of columns such as departure date and time, arrival date and time, total stops, airline, source, destination, and flight fares.

2.7 Data Credibility and Integrity

The dataset used in this project is from a reputable source, and steps are taken to maintain the credibility and integrity of the data throughout the project.

2.8 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn are used to build the whole model.



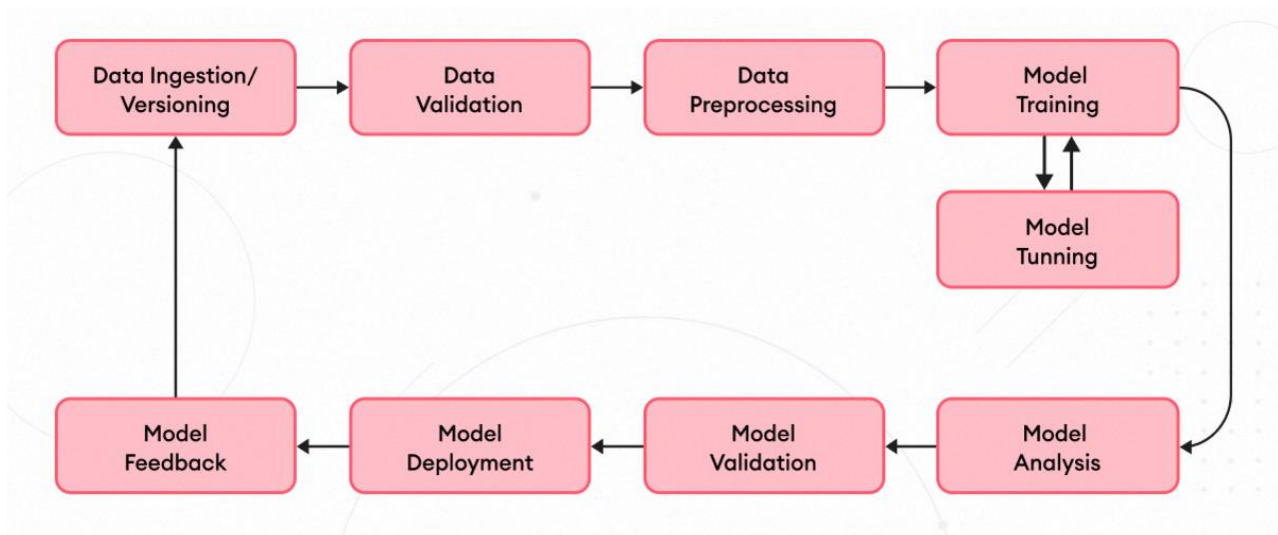
- VS Code is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- Azure is used for deployment of the model.
- Tableau/Power BI is used for dashboard creation.
- Datastax astra is used to retrieve, insert, delete, and update the database.
- Python is used for backend development.
- GitHub is used as version control system.

3 Design Details

3.1 Process Flow

In this phase, feature selection techniques such as correlation matrix, heatmap, Extra Trees Regressor, and Recursive Feature Elimination with Cross-Validation (RFEVCV) were used to select the most important features and for prediction will be using Machine Based models. Below is the process flow diagram is as shown below.

3.1.1 Model Training and Evaluation



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

If errors encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.4 Performance

From the performance of various models, the Random Forest Regressor algorithm was chosen as the final model and hyperparameter tuning was performed using Randomized Search CV for improved performance.

3.5 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.6 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.8 Deployment



4 Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

5 Conclusion

The developed model can accurately estimate the flight fares based on the given features which will help travelers plan their trips more effectively and make informed decisions about flight bookings

