Flight Fare Price Prediction - HLD

Date: 25/10/2023

Team members:
- Yash Keshari
- Md Ehsanul Haque Kanan

Document Version Control

Date Issued	Version	Description	Author
10/10/2023	1	Initial HLD - V 0.1	Yash Keshari
24/10/2023	2	Major changes – V 0.2	Yash Keshari

Contents

Abstract

The Flight Fare Price Prediction project is a machine learning-based solution designed to estimate flight fares with accuracy. By utilizing input features such as airline, source, destination, total stops, and duration, the project's predictive model provides users with fare predictions. Leveraging data ingestion, preprocessing, and model training pipelines, this project empowers users to make informed decisions when booking flights. With a focus on accuracy and responsiveness, this project aims to enhance the user experience and facilitate cost-effective air travel.

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.

2 General Description

2.1 Product Perspective

The FFP based Ticket Price Prediction system is a Machine learning-based Prediction model which will help us to make Price Prediction and take informed decision.

2.2 Problem statement

To create an AI solution for Flight Fare Price Prediction and to implement the following use cases:

- Predict flight fares accurately based on input features such as airline, source, destination, total stops, and duration.
- Assist users in making informed decisions when booking flights by providing fare estimates.

2.3 Proposed solution

The proposed solution for Flight Fare Price Prediction is a machine learning-based system that utilizes historical data and features such as airline, source, destination, total stops, and duration to predict flight fares accurately. This system aims to provide users with reliable fare estimates, facilitating cost-effective air travel planning. The process involves data ingestion, preprocessing, and model training to ensure the accuracy of predictions.

2.4 FURTHER IMPROVEMENTS

The Flight Fare Price Prediction project can be extended with several enhancements:

- Integration of real-time data sources to ensure the most up-to-date fare predictions.

- Implementation of advanced models and algorithms for even more precise fare estimates.
- Integration with weather data for taking weather-related factors into account.
- Integration with airline booking platforms for a seamless booking experience.

2.5 Technical Requirements

To build a robust Flight Fare Price Prediction system, the following technical requirements are necessary:

- Data Management:
  Data ingestion and preprocessing.
  Feature engineering for relevant flight attributes.
  Utilizing a variety of data sources.

- Machine Learning Models:
  Selection and implementation of machine learning algorithms.
  Regular model training and updating.

- User Interface:
  User-friendly interface for inputting flight details and viewing fare predictions.
  Real-time Data Integration:
  Integration of real-time data sources to provide current fare estimates.

- Scalability:
  System scalability to accommodate a high volume of users and requests.

- Quality Assurance:
  Testing procedures to ensure accurate fare predictions.
- Security:
  Implementation of data security measures for user privacy.
- Documentation:
  Comprehensive documentation for users and developers.
- Deployment:
  Setup of deployment processes for user accessibility.
- Performance Monitoring:
  Monitoring system performance and addressing issues proactively.

2.6 Data Requirements

The data requirements for the Flight Fare Price Prediction project align with the nature of the problem statement. They include the following:

- Data Type: The project requires structured data, particularly related to flight details, historical pricing, and influencing factors.

- Data Volume: A sufficient volume of data is necessary, with a diverse dataset that covers various airlines, routes, and flight types. A minimum dataset should include several thousand data points for meaningful predictions.

- Data Annotation: Proper annotation is crucial, with labels or attributes that define the target variable (fare price) and other relevant features.

- Data Format: Data should be organized in a structured format, such as CSV (Comma-Separated Values), which facilitates data preprocessing and modeling.

- Data Range: Ensure data covers a wide range of fare prices, stops, durations, sources, and destinations to capture the full spectrum of

flight options.

- Data Quality: High data quality is essential, with minimal missing values or outliers. Data cleaning and quality control processes should be in place.

- Real-Time Data: To provide current fare predictions, the system should integrate real-time data sources, such as fuel costs, market trends, and seasonal factors.

These data requirements are fundamental to creating a reliable Flight Fare Price Prediction system that can offer valuable fare estimates to travelers.

2.7 Tools used

Tools and Technologies Used -

- **Programming Language**: Python
- **Python Libraries and Frameworks**:
  - NumPy
  - Pandas
  - Scikit-learn

- **Integrated Development Environment (IDE)**: Vs-code
- **Data Visualization**:
  - Matplotlib
  - Seaborn

- **Cloud Computing and Deployment**:
  - AWS (Amazon Web Services)

- **Front-end Development**:
  - Streamlit.
- **GitHub is used as version control**


2.8 Hardware Requirements:

- Laptop
- A stable Internet Connection

2.9 Constraints:

The Flight Fare Price Prediction system should adhere to the following constraints:

- **User-Friendliness**: The system should be designed to be user-friendly, with an intuitive interface that requires minimal technical knowledge from users. The workings of the system should be transparent to users.
- **Automation**: The system should be as automated as possible, reducing the need for manual intervention and ensuring that users can obtain fare predictions effortlessly.

3 Design Details

3.1 Process Flow

Flight Fare Price Prediction Process Flow:

- **Data Collection**:
  Gather flight-related data from various sources, including airlines, routes, historical pricing, and influencing factors.

- **Data Preprocessing**:
  Clean and prepare the data, handle missing values, and transform features for model training.

- **Machine Learning Model Development:**
  Select and implement machine learning algorithms to build a predictive model. Train the model on the preprocessed data to establish the relationship between input features and fare prices.

- **Model Evaluation:**
  Assess the model's performance using metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) and R2 score.

- **User Interface:**
  Design and develop a user-friendly interface, such as a web or mobile app, for users to input their flight details.

- **Real-time Data Integration:**
  Integrate real-time data sources, such as fuel costs and market trends, to provide up-t
  date fare predictions.

- **Model Deployment:**
  Deploy the trained model on a cloud platform (e.g., AWS) to make it accessible to users.

- **User Interaction:**
  Users input their flight details (airline, source, destination, stops, duration) via

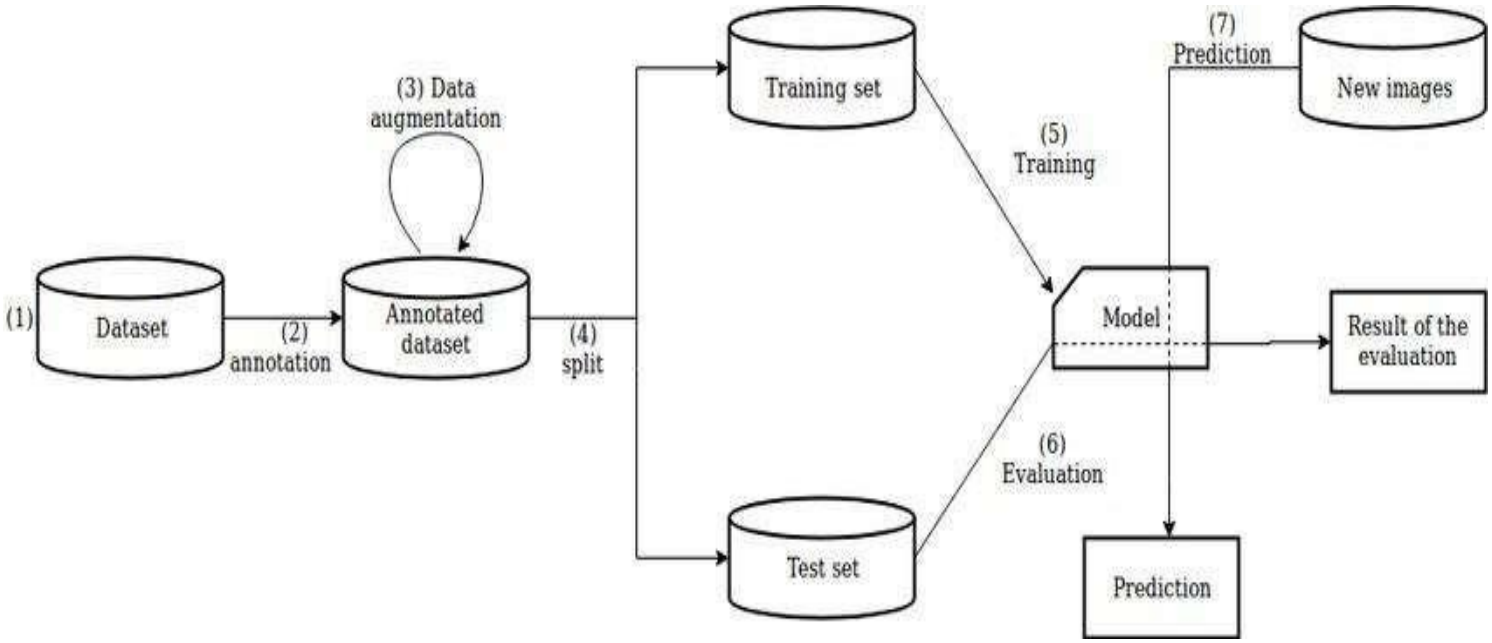
the user interface.

- **Prediction Generation:**
  The system processes the user input through the trained model to generate fare predictions.

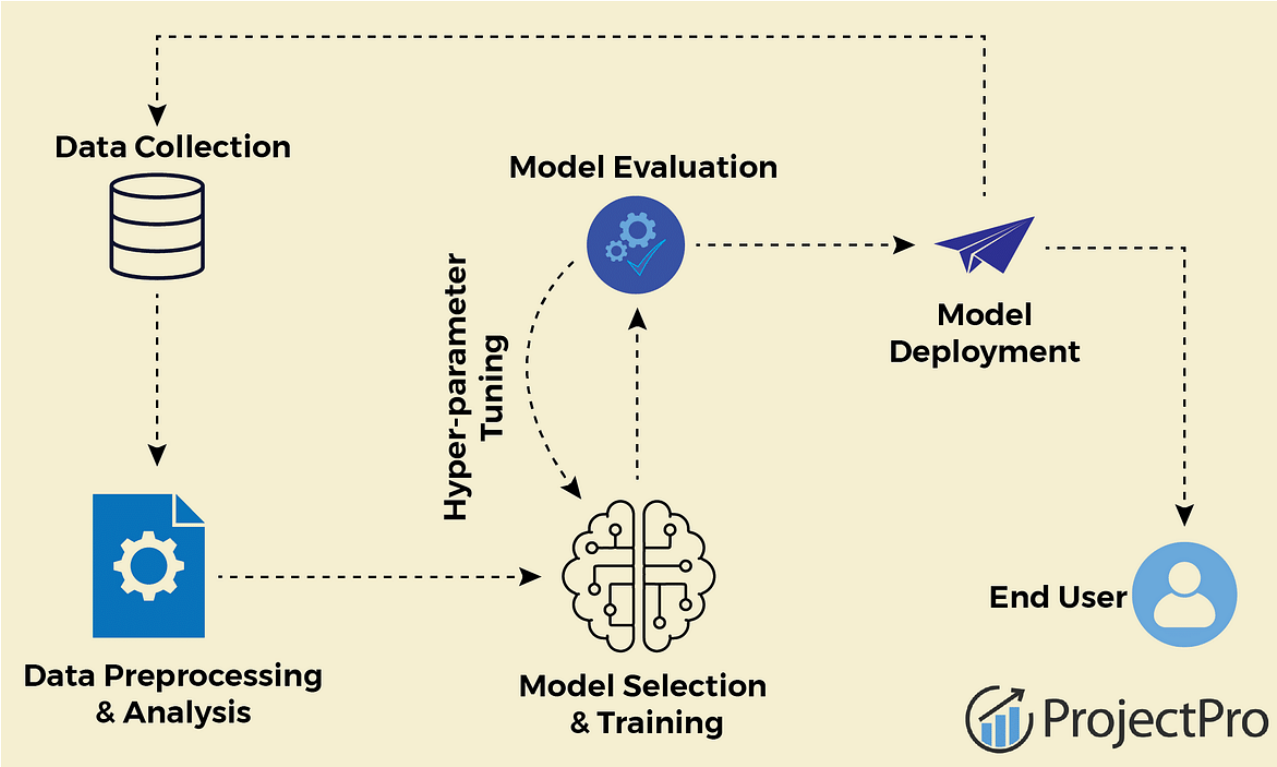
- **Output Presentation:**

Display the fare estimate to the user, allowing them to make informed booking decisions.

3.1.1 Model Training and Evaluation



3.1.1 Deployment Process



3.2 Event log

The system should log every event so that the user will know what process is running internally. This is achieved in 'Logger.py' file in 'Src' folder of our project.

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

Should errors be 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.

This is achieved in 'Exception.py' file inside the 'src' folder of our project.

4 Performance

The Flight Fare Price Prediction system should prioritize high accuracy in estimating flight fares based on various input features, such as airline, source, destination, stops, and duration. Precise predictions are crucial to assist travelers in making informed booking decisions.

- Reusability

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

- 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.

- Resource Utilization

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

- Deployment

The project is deployed on AWS EC2 instance with the API Endpoint .

5 Conclusion

The Flight Fare Price Prediction system is designed to predict the flight fare based on user input. By utilizing historical fare data and machine learning algorithms, the project aims to provide users with the means to know and understand fluctuations in flight prices at an early stage. This information empowers travelers to make timely and cost-effective booking decisions, ensuring a pleasant and budget-conscious travel experience

6 References

- Google images
- Pw Skills documentations
- https://www.projectpro.io/