**High Level Design**

**Flight Price Prediction**

|  |  |
| --- | --- |
| **Written by** | **Denistan B** |
| **Documented Version** | **1.0** |
| **Last Revised Date** | **13 December 2024** |

**Contents**

1. **Abstract**
2. **Introduction**2.1 Why this High-level Document?  
   2.2 Scope  
   2.3 Definitions
3. **General Description**3.1 Product Perspective  
   3.2 Problem Statement  
   3.3 Proposed Solution  
   3.4 Technical Requirements  
   3.5 Dataset  
   3.6 Tools Used
4. **Design Details**4.1 Process Flow  
   4.2 Model Training and Evaluation  
   4.3 Deployment Process
5. **Performance**5.1 Reusability  
   5.2 Comparison of Models  
   5.3 Application Compatibility
6. **KPIs (Key Performance Indicators)**
7. **Conclusion**

## **Abstract**

**In this project, a machine learning model was developed to predict flight fares based on various flight and external features. The dataset includes attributes such as airline, flight duration, time of booking, departure and arrival timings, and the seasonality of travel, among others. Two machine learning models, Linear Regression and Random Forest Regressor, were utilized to estimate flight fares. After training both models on the dataset, their performance was evaluated using R-squared (R²), Mean Squared Error (MSE), and Mean Absolute Error (MAE). The Random Forest Regressor outperformed Linear Regression in all evaluation metrics, achieving a higher R² score and lower MSE and MAE. The final model, Random Forest, was deployed through an API to provide real-time flight fare predictions, assisting travelers in optimizing travel costs and planning efficiently. This solution also holds potential for integration with travel platforms to enhance user experiences.**

## **2. Introduction**

**2.1 Why this High-level Document?**

This document provides a detailed overview of the machine learning model developed for predicting flight fares. It is intended for a technical audience, including data scientists, machine learning practitioners, and industry analysts, to understand the systematic steps taken to build, evaluate, and deploy the model. The document serves as a comprehensive guide, covering key aspects such as data preprocessing, feature engineering, model selection, evaluation metrics, and deployment. Additionally, it outlines the rationale behind the choices made during the project, ensuring transparency and replicability. The insights shared here can also be applied to expand or adapt the model for other predictive use cases in the travel and transportation industry.

**2.2 Scope**

The scope of this document includes:

**1. Data Exploration and Preprocessing:**  
The dataset contains various features related to flights, such as airline name, departure and arrival times, flight duration, ticket class, and external factors like seasonal demand. This step involves:

* Exploring and understanding the dataset.
* Cleaning data by addressing missing values and removing inconsistencies.
* Encoding categorical variables such as airline names and flight classes.
* Performing feature scaling for numeric data to ensure compatibility with machine learning algorithms.
* Analyzing and addressing data imbalances, ensuring all categories are adequately represented.

**2. Model Development:**  
This section describes the machine learning algorithms considered for predicting flight fares. Algorithms include Linear Regression, Decision Trees, and ensemble methods such as Random Forest and Gradient Boosting. It outlines the following:

* Feature engineering to identify the most relevant predictors of flight fares, such as "days to departure" or "seasonal indicators."
* Training the models on historical flight fare data while validating them using cross-validation techniques.

**3. Model Evaluation:**  
The performance of the trained models is evaluated to ensure they generalize well on unseen data. Evaluation involves:

* Using metrics such as Mean Squared Error (MSE), R-squared (R²), and Mean Absolute Error (MAE).
* Comparing model performance to select the best-performing algorithm for deployment.
* Discussing the strengths and weaknesses of each model based on evaluation results.

**4. Deployment Process:**  
After selecting and fine-tuning the best model, it is prepared for deployment. This section includes:

* Saving the trained model in a portable format.
* Setting up APIs to allow real-time or batch access to the model’s predictions.
* Preparing the model for integration with travel booking platforms and user interfaces.  
  The document primarily focuses on model building and technical steps leading up to deployment, excluding real-time monitoring and continuous model updating, which are considered beyond the current scope.

**2.3 Definitions**

**1. Airline:**  
The name of the airline operating the flight. Different airlines may have varying pricing strategies based on reputation, service quality, and operational costs, making this a significant feature in predicting fares.

**2. Flight Duration:**  
The total travel time from the departure to the arrival destination. Longer flights might have higher fares due to increased fuel costs and operational overheads.

**3. Departure Time:**  
The scheduled departure time of the flight. Flights departing during peak hours may be more expensive due to higher demand.

**4. Arrival Time:**  
The scheduled arrival time of the flight. This can also influence fares, as late-night or early-morning arrivals might have lower demand.

**5. Days to Departure:**  
The number of days left until the flight's departure. This is a critical feature, as fares typically increase closer to the departure date due to limited seat availability.

**6. Flight Class:**  
The class of the ticket (e.g., economy, business, first class). Business and first-class tickets generally have higher fares due to premium services offered.

**7. Route:**  
The travel route, including the departure and destination airports. Popular or heavily trafficked routes often show distinct pricing patterns compared to less frequent routes.

**8. Seasonality:**  
Indicates whether the flight is during a peak travel season, such as holidays or festivals. Seasonal variations heavily impact fare trends due to fluctuating demand.

**9. Airline Promotions:**  
Discounts or promotional offers by the airline. These can significantly reduce fares and are an important factor for prediction models.

**10. Ticket Demand:**  
The current demand for tickets on the flight. High-demand flights generally have higher fares as airlines capitalize on the urgency of travelers.

**11. Fare Type (Refundable/Non-Refundable):**  
Indicates whether the ticket is refundable. Refundable tickets are generally priced higher than non-refundable ones due to added flexibility.

**12. Target Variable – Flight Fare:**  
The price of the flight ticket. This is the variable the machine learning model aims to predict based on the other features.

### **3. General Description**

### 3.1 Product Perspective

This predictive model is designed as a tool for travelers and travel service providers to estimate flight fares based on multiple factors. By analyzing airline details, timing, and seasonal trends, the model provides users with insights into fare fluctuations, helping them optimize travel plans and budgets. Travel agencies and booking platforms can integrate this model into their systems to enhance user experiences by offering fare predictions and suggestions for cost-effective bookings.

### 3.2 Problem Statement

The primary challenge is to predict flight fares accurately using features such as airline, flight duration, departure and arrival times, seasonal indicators, and class type. The complexity arises due to the dynamic nature of flight pricing, which depends on numerous factors including demand, availability, and external influences like holidays or events. The objective is to develop a machine learning model that can identify patterns in historical data and leverage these patterns to predict fares for unseen data. This regression task aims to estimate continuous fare values based on input features.

### 3.3 Proposed Solution

To address the problem, the proposed solution involves building a robust regression model capable of accurately predicting flight fares. The development process includes:

* **Data Preprocessing:** Cleaning and preparing the dataset, handling missing values, encoding categorical variables, and normalizing numerical data.
* **Feature Engineering:** Deriving additional features such as "days to departure," "holiday season indicator," or "peak hour flag" to enhance model performance.
* **Model Selection:** Experimenting with different machine learning algorithms, such as Linear Regression, Random Forest Regressor, and Gradient Boosting Machines, to identify the most suitable model.
* **Model Evaluation:** Assessing the model's accuracy using metrics like Mean Squared Error (MSE), R-squared (R²), and Mean Absolute Error (MAE) to ensure it generalizes well to unseen data.
* **Deployment:** Deploying the final model via APIs or as part of a web or mobile application to provide real-time fare predictions for end-users.

### 3.4 Technical Requirements

* **Programming Language:** Python 3.7+
* **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
* **Frameworks and Tools:** Flask/FastAPI for backend, Streamlit for creating a simple user interface

### 3.5 Dataset Overview

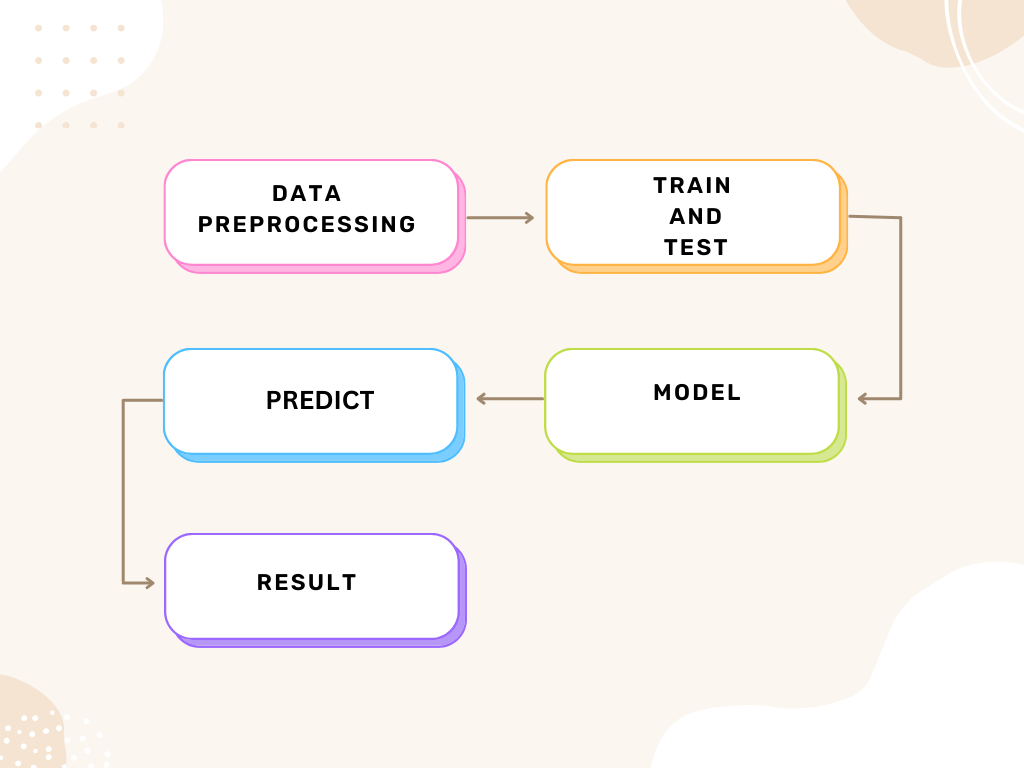
The dataset consists of flight-related features, including:

* **Airline Name**
* **Flight Duration**
* **Departure and Arrival Times**
* **Days to Departure**
* **Seasonal Indicators**
* **Ticket Class**  
  The training dataset contains historical fare information used to train the model, while the test dataset includes only input features, with the goal of predicting fares for these entries.

### 3.6 Tools Used

* **Programming:** Python for development
* **Data Processing and Analysis:** Pandas, NumPy for handling and exploring data
* **Modeling:** Scikit-learn for building and evaluating machine learning models
* **Visualization:** Matplotlib and Seaborn for data visualization and trend analysis
* **Interface Development:** Streamlit or Flask/FastAPI for user-friendly deployment

### **4.Design details**



**4.1 Process Flow**

The process flow for this sales prediction project follows these stages:

1. **Data Collection**: Dataset obtained, containing product and store features, along with sales data for training and testing.
2. **Data Preprocessing**: Cleaning the dataset, handling missing values, encoding categorical variables, and scaling numerical features.
3. **Feature Engineering**: Creating new features, such as calculating store age and performing exploratory data analysis (EDA).
4. **Model Selection**: Evaluating **Linear Regression** and **Random Forest Regressor** models.
5. **Model Training**: Training both models using the training dataset.
6. **Model Evaluation**: Evaluating models using **R-squared**.
7. **Deployment**: Deploying the best-performing model by streamlit.

**4.2 Model Training and Evaluation**

### Linear Regression

**Training:**  
Linear Regression is a simple and interpretable machine learning algorithm used to model the relationship between a dependent variable (target) and one or more independent variables (features). It assumes a linear relationship between the input features and the target variable, fitting a straight line (or hyperplane in multiple dimensions) to minimize the residual sum of squares (RSS) between observed and predicted values. The model parameters are optimized using Ordinary Least Squares (OLS) or Gradient Descent techniques.

**Evaluation:**

* **R-squared (R²):** Measures the proportion of variance in the target variable explained by the model. Linear Regression often provides moderate R² values, as it assumes linearity and may not capture complex relationships in the data.
* **MSE (Mean Squared Error):** Represents the average squared difference between actual and predicted values. Linear Regression generally yields higher MSE compared to advanced models, particularly when the relationships in the data are non-linear.
* **MAE (Mean Absolute Error):** Indicates the average absolute differences between actual and predicted values. Linear Regression's MAE may be higher in datasets with non-linear or complex patterns, as it lacks the flexibility to adapt to such data structures.

### Random Forest Regressor

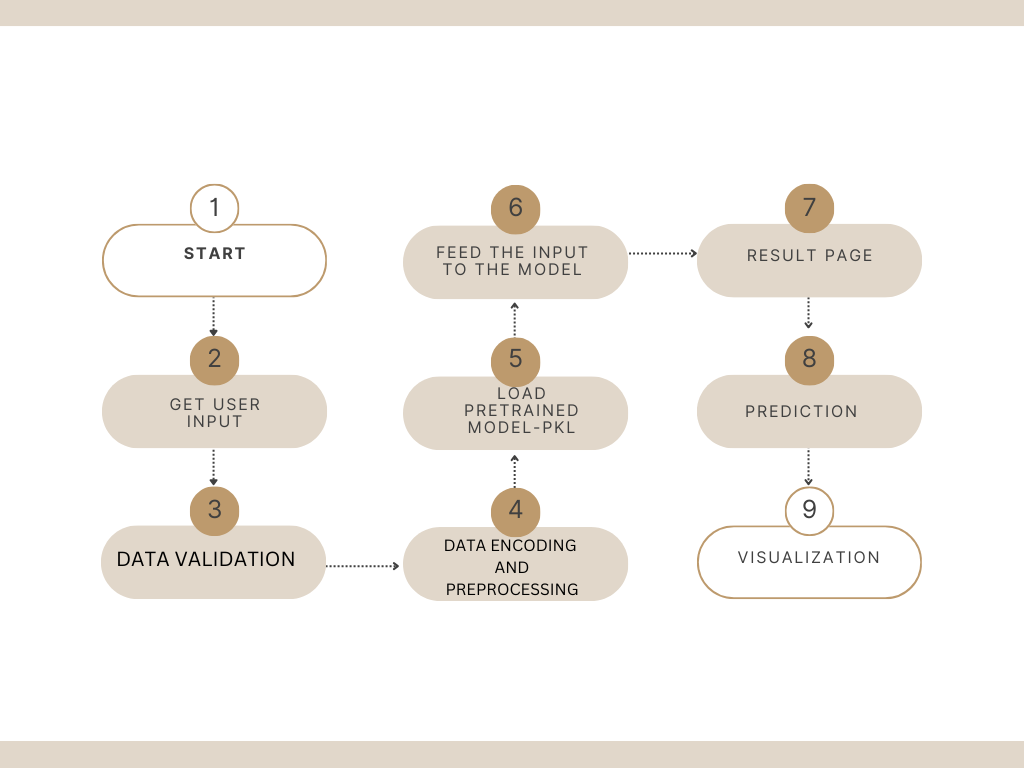
**Training:**  
Random Forest is an ensemble learning technique that builds multiple decision trees during training and aggregates their predictions (by averaging in regression tasks) to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the dataset, and at each split, a random subset of features is considered, which increases model diversity and robustness. This process helps Random Forest effectively handle non-linear relationships and interactions between features.

**Evaluation:**

* **R-squared (R²):** Measures the proportion of variance in the target variable explained by the model. Random Forest typically achieves high R² values due to its ability to model complex, non-linear relationships and interactions between variables.
* **MSE (Mean Squared Error):** Indicates the average squared difference between actual and predicted values. Random Forest generally yields low MSE by leveraging ensemble predictions, which reduce errors from individual decision trees.
* **MAE (Mean Absolute Error):** Represents the average absolute differences between actual and predicted values. Random Forest often produces low MAE by accurately capturing patterns and minimizing prediction errors, even in datasets with significant variability.

|  |  |
| --- | --- |
| **Model** | **R² Score** |
| **Random Forest** | **0.8324** |
| **Polynomial** | **0.0241** |
| **Linear** | **0.5610** |

**4.3 Deployment Process**

****

* Serialize the trained model using Pickle.
* Develop a Streamlit-based web application for user interaction.

### **5. Performance**

### 5.1 Reusability

* The flight fare prediction model is designed to be reusable for datasets with a similar structure. By adapting the data preprocessing pipeline to include airline-specific, seasonal, or route-based features, the model can be applied to other domains, such as train or bus fare predictions, or even dynamic pricing in other industries.
* The modular design of the data pipeline, feature engineering, and training components allows for easy customization for new datasets, reducing the need for complete retraining. Only minimal adjustments, such as re-tuning hyperparameters or updating the feature set, are required.

### 5.2 Comparison of Models

Both **Random Forest Regressor** and **Gradient Boosting Regressor** are effective at handling non-linear relationships and feature interactions in the flight fare dataset. However, their strengths differ:

* **Random Forest Regressor:**
  + Excels in robustness and stability, requiring minimal hyperparameter tuning.
  + Less sensitive to noise due to averaging across multiple decision trees.
  + Suitable for general-purpose tasks where interpretability and simplicity are key priorities.
  + Faster training compared to Gradient Boosting, though predictions may take longer with large datasets.
* **Gradient Boosting Regressor:**
  + Sequentially optimizes weak learners, often achieving higher accuracy on complex datasets.
  + Offers fine-grained control via hyperparameter tuning, though this increases computational requirements.
  + Can sometimes outperform Random Forest in terms of precision, especially when carefully tuned.
* **Conclusion:**  
  While both models deliver strong results, Random Forest is preferred for its simplicity and reliability in broader applications, whereas Gradient Boosting is ideal for tasks demanding high precision and tailored optimization.

### 5.3 Application Compatibility

* The deployed flight fare prediction model is built with compatibility in mind for integration with existing travel booking platforms and user-facing applications.
* **API Integration:** The model provides seamless access to predictions through RESTful APIs, ensuring compatibility with both web-based and mobile applications.
* **Scalability:** It can handle large volumes of prediction requests, making it suitable for travel agencies, online booking systems, and customer apps.
* **Extensibility:** The modular nature of the deployment ensures that additional features, such as user preferences or loyalty discounts, can be incorporated with minimal modifications.
* **Real-time Usability:** The model’s efficient design allows for real-time predictions, providing immediate fare estimates to assist users in making informed booking decisions.

### **6. KPIs (Key Performance Indicators)**

* **Prediction Accuracy (R² Score):** Achieved > 0.83
* **Business Impact**: Measuring how accurate predictions improve inventory and sales forecasting, reducing stockouts and overstock situations.
* **Model Retraining Frequency**: Regular monitoring and retraining of the model to maintain performance as market conditions evolve.

**7.Conclusion**

The **Random Forest Regressor** and **Linear Regressor** models have shown exceptional performance in predicting sales, achieving high accuracy through their ability to capture complex relationships within the data. These models have been successfully deployed for real-time sales predictions, providing valuable insights for inventory management and supporting effective business decision-making in retail systems.

Future enhancements include integrating external factors such as seasonal trends and economic indicators, enabling automated updates to the models, and adopting cloud-based solutions for greater scalability. Further, incorporating advanced visualizations and exploring hybrid ensemble techniques can refine predictions and offer deeper insights into sales behavior, ensuring the system continues to meet dynamic business requirements effectively.