Detail Project Report

Flight Fare Prediction

**Contents**

Document Version Control **……………………………………………………………………….2**

Abstract **…………………………………………………………………………………………...4**

1. Introduction **…………………………………………………………………………………...5**

1.1 Why this DPR Document?**................……………………………………………………...5**

1. General Description**……………………………..…………………………………..………...6**

2.1 Problem Perspective**………………………….…………………………………………...6**

2.2 Problem Statement**………………………………………………………………………...6**

2.3 Proposed Solution**…………………………………….…………………………………...6**

2.4 Further Improvements**……………………………………………………..……………...6**

1. Technical Requirements **………………………………………………………………………7**

3.1 Tools Used**……………………………………….………………………………………...7**

1. Data Requirements **…………………………………………………………………………….8**
   1. Data Collection**……………………………………………..……………………………...8**
   2. Data Description**…………………………………………………………………………...8**
   3. Importing Data into Database**………………………………………………..….................8**
   4. Exporting Data from Database**……………….……………………………………………8**

4.5 Data Preprocessing**………………………………………………………………………...8**

1. Design Flow **……………………………………………………………………………………….9**
   1. Modelling Creation and Evaluation**………………………………………….………........9**
   2. UI Integration**……………...…………………….………………………………………...9**
   3. Deployment Process**……………………………………………………….........................9**
   4. Logging **…………………………………………………………………………………...9**
   5. Data from User**……………………………………………………….................................9**
   6. Data Validation**…………………………………………………………………………...10**
   7. Rendering the Results**…………………………………………………………………….10**
   8. Deployment**……………………………………………………………………………....10**
2. Conclusion **…………………………………………………………………………….………….11**
3. Frequently Asked Questions (FAQs) **……………………………………………….…………….12**
4. **Abstract**
5. The recent changes in the international market have had a significant impact on the aviation sector due to various reasons. This impact affects both business and customer perspectives. The major reason for such an impact is the implementation of different rules by governments for various airline companies worldwide. As a result, the prices of flight tickets have varied from one place to another. Booking a flight ticket involves different criteria that determine the price, such as online bookings and offline bookings. In this machine learning implementation, we will explore the factors that impact the price of flight tickets and predict the appropriate price.

**1. Introduction**

**1.1 Why this DPR Document?**

The main purpose of this DPR documentation is to provide the necessary details of the project and describe the machine learning model and written code. It also offers a detailed description of how the entire project has been designed end-to-end.

Key Points:

* Describes the design flow
* Implementations
* Software requirements
* Architecture of the project
* Non-functional attributes such as reusability, portability, and resource utilization

**2 General Description**

**2.1 Problem Perspective**

The flight fare prediction is a machine learning model that helps users predict the price of flight tickets and understand the price of their journey.

**2.2 Problem Statement**

After the amendment of new rules, there are changes in the flight fare price from one location to another. The main goal of the system is to create a model to predict the price of flight fare based on user input, such as the date of the journey, source, destination, and more.

**2.3 Proposed Solution**

To solve the problem, we have created a user interface for taking input from the user to predict the flight fare price using our trained ML model. After processing the input, the predicted value from the model is communicated to the user.

**2.4 Further Improvements**

We also analyze the data used for training the ML model by considering different occasions, such as weekday, season, or any social reasons, and considering different business angles. If we use such information and predict the discounted flight fare price, it will bring some loss to the airline companies but benefit the users. If we develop these using the business perspective of airlines, this technique isn't thought-about.

**3. Technical Requirements**

As technical requirements, we don't need any specialized hardware for the virtualization of the application. The user should have a device that has access to the web and a fundamental understanding of providing input.

**3.1 Tools Used**

* Python 3.9: Programming language and frameworks like NumPy, Pandas, Scikit-learn, and other modules for building the model.
* Jupyter Notebook: Used as an IDE.
* Data Visualizations: Seaborn and components of Matplotlib.
* Data Collection: Using Prophetess info.
* Front-End Development: HTML/CSS.
* Backend Deployment: Flask.
* Version Management: GitHub.
* Deployment: AWS

**4. Data Requirements**

The data requirements are based on the problem statement, and the dataset is accessible on Kaggle in the file format of .xlsx.

**4.1 Data Collection**

The data for this project is collected from the Kaggle Dataset. The URL for the dataset is [kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh](https://kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh).

**4.2 Data Description**

Flight Fare Prediction is a dataset containing over 10,000 entries, publicly available on Kaggle. The information in the dataset is present in two separate Excel files named train.xlsx and test.xlsx. The dataset includes information such as Date of Journey, Source, Destination, Arrival Time, Departure Time, Total Stops, Airlines, Additional Info, and Price.

**4.3 Importing Data into Database**

An API is created for the transfer of the data into the Cassandra database. The steps performed are as follows:

* Establish a connection with the database.
* Create a database named FlightInfo.
* Use the cqlsh command to create the necessary table in the database.
* Finally, use a cqlsh command to bulk insert the dataset into the table.

**4.4 Exporting Data from Database**

The API also provides a download URL, which allows users to export the data from the database to a CSV file.

**5. Design Flow**

**5.1 Modelling Creation and Evaluation:**

For the Flight Fare Prediction project, we utilized the Random Forest Regressor algorithm to model our data and predict flight fares. The dataset was preprocessed and split into training and testing data. The Random Forest Regressor was trained on the training data, and its performance was evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). We also analyzed the model's feature importances to understand which factors significantly impact the flight fare prediction.

**5.2 UI Integration:**

To create a user-friendly interface for the Flight Fare Prediction project, we used HTML and CSS for front-end development. Flask, a Python web framework, was employed for both data and backend preparation. We integrated the machine learning model with the HTML web page to allow users to input their flight details and receive fare predictions.

**5.3 Deployment Process**:

The deployment process involved using the Heroku platform to host the project. The tested machine learning model was deployed on Heroku, enabling users to access the application from any internet-enabled device. Heroku's cloud service allowed us to run the application seamlessly and provided a user-friendly experience for accessing the flight fare prediction system.

**5.4 Logging:**

During the development and deployment stages, we implemented logging to track system events, errors, and exceptions. Whenever an error or exception occurred, the system logged the event with relevant details such as reason and timestamp. This logging mechanism facilitated easy debugging and helped us rectify any issues that arose during the project's lifecycle.

**5.5 Data from User:**

Data from the user was collected through the HTML web page created for the project's user interface. Users provided information like the date of journey, source, destination, and other relevant details needed for flight fare prediction.

**5.6 Data Validation:**

After retrieving data from the user, it was processed by the app.py file and underwent validation. The validation process ensured that the provided data was correct and met the required format. Validated data was then sent to the machine learning model for flight fare prediction.

**5.7 Rendering the Results:**

The data sent for prediction was processed by the machine learning model, and the predicted flight fare was rendered back to the web page. Users could view the approximate flight fare for their journey based on the input they provided.

**5.8 Deployment:**

After thorough testing and validation, the machine learning model was deployed on the Heroku platform. The deployment enabled users to access the Flight Fare Prediction system using their internet-enabled devices, making it easily accessible and user-friendly.

**6. Conclusion**

The Flight Fare Prediction system has been developed to assist customers with accurate price predictions using a trained machine learning model and a set of rules. By inputting relevant details such as the date of the journey, source, destination, and other factors, users can obtain an approximate value of their flight fare. The system takes into consideration various factors that impact flight ticket prices and uses this information to provide reliable predictions.

Customers can benefit from this system as it helps them plan and budget for their travel expenses more effectively. By gaining insights into the price of flight tickets, users can make informed decisions and choose the most suitable options for their journey. The system's predictions are based on the analysis of historical flight fare data and trained machine learning algorithms, ensuring a reliable and accurate prediction process.

Overall, the Flight Fare Prediction system aims to empower customers with the knowledge they need to make cost-effective travel decisions, enhancing their overall travel experience.

**7. Frequently Asked Questions (FAQs)**

**Q1) What's the source of data?**

The data for training is provided by the client in multiple batches, and each batch contains multiple files.

**Q2) What was the type of data?**

The data was a combination of numerical and categorical values.

**Q3) What's the complete flow you followed in this Project?**

Refer to Page no 9 for better understanding.

**Q4) After the File validation, what do you do with incompatible files or files which didn't pass the validation?**

Files like these are moved to the Archive Folder, and a list of these files has been shared with the client, and we removed the bad data folder.

**Q5) How logs are managed?**

We are using different logs as per the steps that we follow in validation and modeling, like File validation log, Data Insertion, Model Training log, prediction log, etc.

**Q6) What techniques were you using for data pre-processing?**

* Removing unwanted attributes.
* Visualizing the relation of independent variables with each other and output variables.
* Checking and changing the distribution of continuous values.
* Removing outliers.
* Cleaning data and imputing if null values are present.
* Converting categorical data into numeric values.

**Q7) How was the training done, or what models were used?**

* Before dividing the data into training and validation sets, we performed pre-processing over the dataset and made the final dataset.
* As per the dataset, training and validation data were divided.
* Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used. Based on the recall, the final model was used on the dataset, and we saved that model.

**Q8) How was Prediction done?**

The testing files are shared by the client. We performed the same life cycle on the provided dataset. Then, based on the dataset, the model is loaded, and prediction is performed. In the end, we get the accumulated data of predictions.

**Q9) What are the different stages of deployment?**

* First, the scripts are stored on GitHub as a storage interface.
* The model is first tested in the local environment.
* After successful testing, it is deployed on AWS

This document represents the Detailed Project Report for the Flight Fare Prediction project. It provides an overview of the project's technical requirements, data requirements, design flow, conclusion, and frequently asked questions. The project aims to predict flight fare prices based on various inputs provided by users. It utilizes machine learning algorithms and data preprocessing techniques for accurate predictions.