Detail Project Report (DPR)

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
0.2	11-06-2023	Rajesh	Deployment and Process
0.3	14-06-2023	Rajesh	Q and A

Contents

1. Introduction

1.1. What is High-Level design document?

The main purpose of this HLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

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. 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 goal of the machine learning project is to create a programme that predicts the fare of the flight by taking bound input from the user like date and time of journey, source and destination, airline company, number stops between the journey etc.

2.3. Purposed Solution

The Airline Flight Fare Prediction is a Flask web application to predict airline flight fares across the Indian cities. The dataset for the project is taken from Kaggle, and it is a timestamped dataset so, while building the model, extensive pre-processing was done on the dataset especially on the date-time columns to finally come up with a ML model which could effectively predict airline fares across various Indian Cities. The dataset had many features which had to pre-processed and transformed into new parameters for a cleaner and simple web application layout to predict the fares.

The various independent features in the dataset were:

Airline: The name of the airline.
Date_of_Journey: The date of the journey
Source: The source from which the service begins.

Destination: The destination where the service ends.
Route: The route taken by the flight to reach the destination.
Dep_Time: The time when the journey starts from the source.
Arrival_Time: Time of arrival at the destination.
Duration: Total duration of the flight.
Total_Stops: Total stops between the source and destination.
Additional_Info: Additional information about the flight
Price: The price of the ticket

We have used the regression algorithms to find out the expected price of the flight for the given set of input. We are using the sample dataset from the Kaggle to develop the model and created both instance and batch prediction. In instance prediction, user can provide input on the HTML webpage and on click of the predict button, the user will get to know the predicted fare of the flight for the given set of inputs. In batch prediction, user can save .xlsx file in the Input_folder and can find the predicted prices in the Output_folder.

2.4. Technical Requirements

The user can access the webpage and should have the fundamental understanding of providing the input. And rest AWS itself will take care the backend requirements to run all the package that are needed for the process the provided information and to show the results..

2.5. Data Requirements

The info demand is totally supported the matter statement. and also, the information set is accessible on the Kaggle within the type of standout sheet(.xlsx). because the main theme of the project is to induce the expertise of real time issues, we have a tendency to ar once more mercantilism {the information into the prophetess data base and commerce it into csv format.

2.6. Tool Used

- Python 3.8 is used while creating the environment.
- VS Code is used as IDE.
- AWS is used for deployment.
- HTML is used for developing the webpage for the instance prediction.
- GitHub is used as code repository.

2.7. Data Gathering

The data for the current project is being gathered from Kaggle dataset, the link to the data is: https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh

2.8. Data Description

There are about 10k+ records of flight information such as airlines, data of journey, source, destination, departure time, arrival time, duration, total stops, additional information, and price. A glance of the dataset is shown below.

	Airline	e_of_Jour	Source	Destinatior	Route	Dep_Time	rrival_Tim	Duration	otal_Stop	ditional_I	Price
2	IndiGo	24/03/201	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 M	2h 50m	non-stop	No info	3897
3	Air India	1/05/2019	Kolkata	Banglore	CCU → IXF	05:50	13:15	7h 25m	2 stops	No info	7662
4	Jet Airway	9/06/2019	Delhi	Cochin	DEL → LKC	09:25	04:25 10 J	19h	2 stops	No info	13882
5	IndiGo	12/05/201	Kolkata	Banglore	CCU → NA	18:05	23:30	5h 25m	1 stop	No info	6218
6	IndiGo	01/03/201	Banglore	New Delhi	BLR → NA	16:50	21:35	4h 45m	1 stop	No info	13302
7	SpiceJet	24/06/201	Kolkata	Banglore	CCU → BLI	09:00	11:25	2h 25m	non-stop	No info	3873
8	Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	18:55	10:25 13 M	15h 30m	1 stop	In-flight m	11087
9	Jet Airway	01/03/201	Banglore	New Delhi	BLR → BOI	08:00	05:05 02 M	21h 5m	1 stop	No info	22270
10	Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	08:55	10:25 13 M	25h 30m	1 stop	In-flight m	11087
11	Multiple c	27/05/201	Delhi	Cochin	DEL → BO	11:25	19:15	7h 50m	1 stop	No info	8625
12	Air India	1/06/2019	Delhi	Cochin	DEL → BLF	09:45	23:00	13h 15m	1 stop	No info	8907
13	IndiGo	18/04/201	Kolkata	Banglore	CCU → BLI	20:20	22:55	2h 35m	non-stop	No info	4174
14	Air India	24/06/201	Chennai	Kolkata	MAA → CC	11:40	13:55	2h 15m	non-stop	No info	4667
15	Jet Airway	9/05/2019	Kolkata	Banglore	CCU → BO	21:10	09:20 10 M	12h 10m	1 stop	In-flight m	9663
16	IndiGo	24/04/201	Kolkata	Banglore	CCU → BLI	17:15	19:50	2h 35m	non-stop	No info	4804
17	Air India	3/03/2019	Delhi	Cochin	DEL → AM	16:40	19:15 04 M	26h 35m	2 stops	No info	14011
18	SpiceJet	15/04/201	Delhi	Cochin	DEL → PN	08:45	13:15	4h 30m	1 stop	No info	5830
19	Jet Airway	12/06/201	Delhi	Cochin	DEL → BO	14:00	12:35 13 J	22h 35m	1 stop	In-flight m	10262

3. Data Pre-processing

- Initiating the pr-processing by removing the missing rows from the data.
- Removing the duplicate records from the data
- Splitting date, time and duration features and converting them into integers.
- Encoding the categorical data into integers using respective dictionaries.
- Saving the dictionaries to encode the input values during prediction.

4. Design Flow

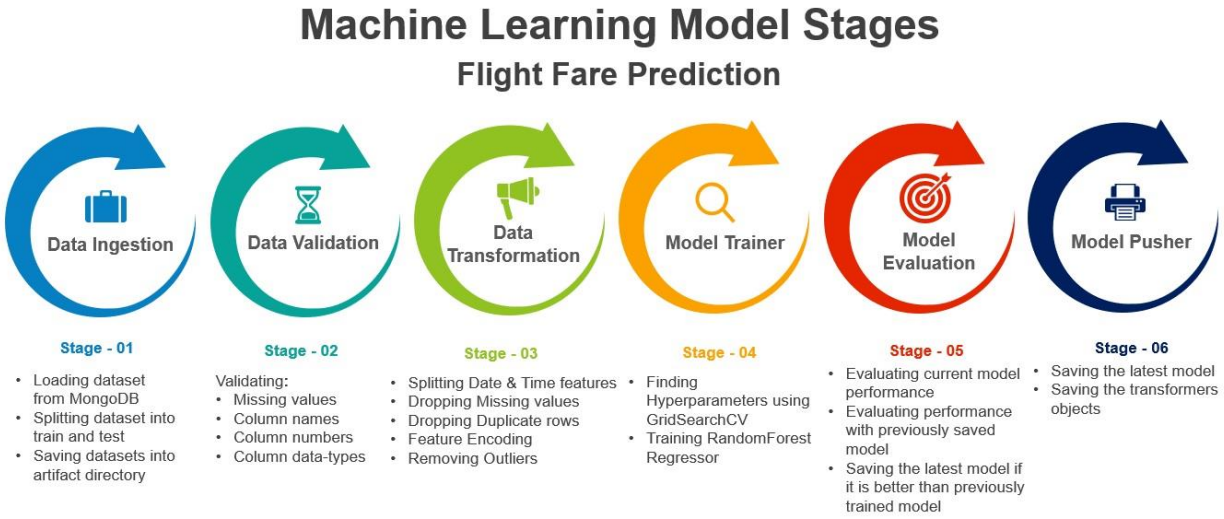
4.1. Modelling

Pre-processed data has been passed to various machine learning regression models along with their hyper-parameters provided by GridSearchCV. The RandomForest Regressor outperform among the given regression algorithms.

Currently trained model is evaluated with the previously saved trained model, if available. If currently trained model is better than the previously trained model then the current model and transformer objects are pushed and saved for the future use.

Both the instance and batch prediction can be performed using the code. The app.py file can be used for instance prediction by taking input values from the user through HTML page and the main.py file can be used batch prediction or training the model as per the requirement.

4.2. Modelling and Deployment Process



1.1. Data from User

The data from the user is retrieved from the created HTML web page.

1.2. Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

1.3. Rendering Result

The data sent for the prediction is then rendered to the web page.

1.4. Deployment

AWS's EC2 and ECR are used to help us to deploy the instance prediction model.

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.

6. Q & A

Q1) What's the source of data?

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

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Refer Page no 6 for better Understanding.

Q 4) After the File validation what you do with incompatible file or files which didn't pass the validation?

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

Q 5) How logs are managed?

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

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking and changing Distribution of continuous values

- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

Q 7) How training was done or what models were used?

- Before dividing the data in training and validation set, we performed pre-processing over the data set 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, final model was used on the dataset and we saved that model.

Q 8) How Prediction was done?

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

Q 9) 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 EC2 machine.