DPR

FLIGHT FARE PREDICTION

Revision Number – 1.3

Last Date of Revision – 9/1/2022

Somay , Madhu

Document Version Control

Date	Version	Description	Author
7-1-2022	1.0	Abstract, Introduction	Somay
8-1-2022	1.1	Deployment and Process	Somay
9-1-2022	1.2	Q and A	Somay

Contents

Abstract

The recent global situations had a huge impact on the aviation sector due to many reasons. This impact has two category people, the first is business perspective and the second is the customers perspective. As safety is the major reason for such impact on the aviation sector, the governments around the world amended different rules to their respective airlines companies. These restrictions had made the availability of the flights and their attendee capacity less. Taking all these factors in consideration the cost of the flight tickets has increased and vary from one place to the other. Booking a flight ticket has split into two, one is the online and the other is the offline bookings. Both these have their respective criteria for cost of the ticket, one such example is the server load and the number of booking requests. In this machine learning implementation, we will see various factors that impact the cost of the flight ticket and predict the appropriate price of the ticket.

INTRODUCTION

Why this DPR Documentation?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on 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 like:
    - Reusability
    - Portability
    - Resource utilization

1 Description

1.1 Problem Perspective

The flight fare prediction is a machine learning model which helps us to predict the cost of the flight ticket and helps the users to know the cost of their journey.

1.2 Problem Statement

The main goal of the project is to create a user interface which provides the cost of the flight ticket by taking certain input from the user like date of journey, onboard location and destination etc.

1.3 Proposed Solution

The solution proposed to take the required input of user from the created interface and process all the provided data to meet the requirements of the machine learning model and finally display the output saying so and so amount is the predicted cost.

1.4 Solution Improvements

We can even predict the cost of ticket considering whether is it a weekday, holiday season or other social reasons. But considering from the perspective of business, if we process such data and predict the cost of the discounted ticket it will bring some loss to the airlines company. Hence this method is not considered.

2 Technical Requirements

There are no hardware requirements required for using this application, the user must have an interactive device which has access to the internet and must have the basic understanding of providing the input. And for the backend part the server must run all the software that is required for the processing the provided data and to display the results.

2.1 Tools Used

- Python 3.9 is used as the programming language and frame works like numpy, pandas, sklearn and other modules for building the model.
- PyCharm is used as IDE.
- For visualizations seaborn and parts of matplotlib are being used.
- For data collection Cassandra database is being used.
- Front end development is done using HTML/CSS.
- Flask is used for both data and backend deployment.
- GitHub is used for version control.
- Heroku is used for deployment.

3 Data Requirements

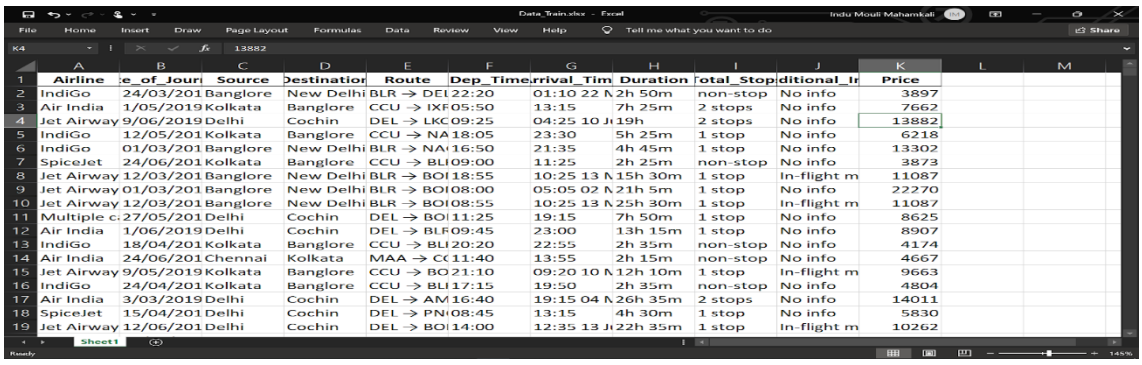
The data requirement is completely based on the problem statement. And the data set is available on the Kaggle in the form of excel sheet(.xlsx). As the main theme of the project is to get the experience of real time problems, we are again importing the data into the Cassandra data base and exporting it into csv format.

3.1 Data Gathering from Main Source

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

3.2 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:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Airline	e_of_Jour	Source	Destinatior	Route	Dep_Time	rrival_Tim	Duration	otal_Stop	ditional_In	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 → IXR	05:50	13:15	7h 25m	2 stops	No info	7662		
4	Jet Airway	9/06/2019	Delhi	Cochin	DEL → LKO	09:25	04:25 10 Ju	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 → BO	18:55	10:25 13 M	15h 30m	1 stop	In-flight m	11087		
9	Jet Airway	01/03/201	Banglore	New Delhi	BLR → BO	08:00	05:05 02 M	21h 5m	1 stop	No info	22270		
10	Jet Airway	12/03/201	Banglore	New Delhi	BLR → BO	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 → BLR	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 Ju	22h 35m	1 stop	In-flight m	10262		

3.3 Import Data into Database

Created an api for the upload of the data into the Cassandra database, steps performed are:

- Connection is made with the database.
- Created a database with name flightfare.
- Cqlsh command is written for creating the data table with required parameters.
- And finally, a cqlsh command is written for uploading the dataset into data table by bulk insertion.

3.4 Export Data from Database

In the above created api, the download url is also being created, which downloads the data into a csv file format.

4 Data Pre-Processing

Steps performed in pre-processing are:

- First the data types are being checked and found only the price column is of type integer.
- Checked for null values as there are few null values, those rows are dropped.
- Converted all the required column into the date time format.
- Performed one-hot encoding for the required columns.
- Scaling is performed for required data.

And, the data is ready for passing to the machine learning algorithm.

5 Design Flow

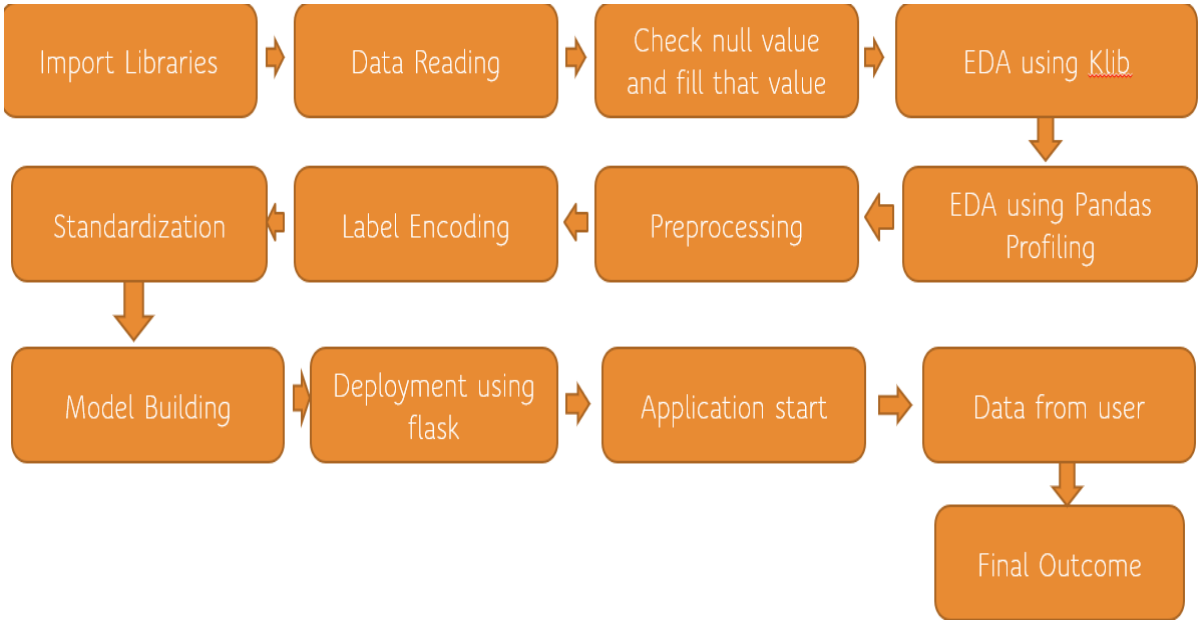
5.1 Modelling

The pre-processed data is then visualized and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modelling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well and further hyperparameter tuning is done to increase the model's accuracy.

5.2 UI Integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

5.3 Modelling Process&5.4 Deployment Process



6 Data from User

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

7 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.

8 Rendering the Results

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

9 Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.

Conclusion

The flight fare prediction can predict the price based on the trained data set in the algorithm. Hence the user can know the approximate cost for their journey.

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 modeling 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 Heroku.