

Architecture

FLIGHT FARE PREDICTION

Revision Number – 1.0

Last Date of Revision – 29-04-2023

Sagar Durgude

Document Version Control

Date	Version	Description	Author
26-04-2023	1.0	Abstract	Sagar
28-04-2023	1.1	Architecture	Sagar
29-04-2023	1.2	Architecture Design	Sagar

Contents

<u>Abstract</u>	<u>4</u>
<u>Introduction</u>	<u>4</u>
<u>Why this Architecture Design documentation?</u>	<u>4</u>
<u>1 Architecture</u>	<u>4</u>
<u>2 Architecture design</u>	<u>5</u>
<u>2.1 Data gathering from main source</u>	<u>5</u>
<u>2.2 Data description</u>	<u>5</u>
<u>2.3 Upload data into Cassandra</u>	<u>5</u>
<u>2.4 Export data from database</u>	<u>5</u>
<u>2.5 Data pre-processing</u>	<u>5</u>
<u>2.6 Modelling</u>	<u>6</u>
<u>2.7 UI integration</u>	<u>6</u>
<u>2.8 Data from user</u>	<u>6</u>
<u>2.9 Data validation</u>	<u>6</u>
<u>2.10 Rendering the results</u>	<u>6</u>
<u>2.11 Deployment</u>	<u>-</u>

Abstract

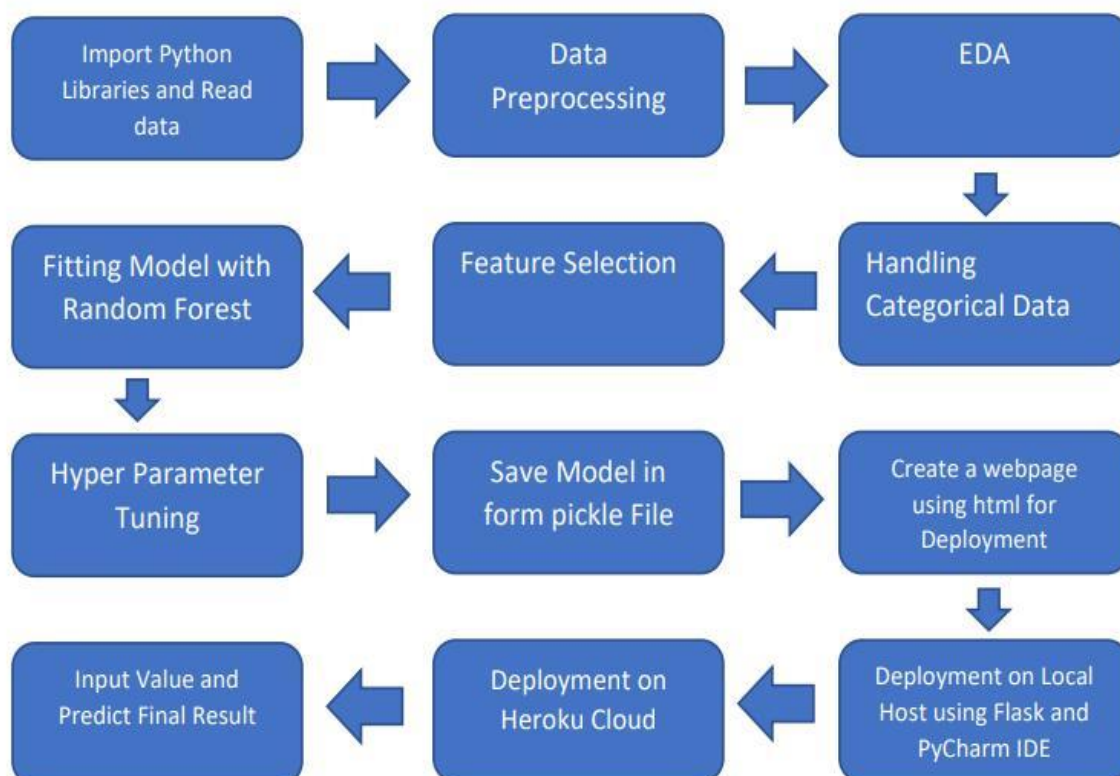
The recent global situations had a huge impact on the aviation sector due to many reasons. This impact has two category people, the first is business perspective and the second is the customers perspective. As safety is the major reason for such impact on the aviation sector, the governments around the 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 fare of the ticket.

Introduction

Why this Architecture Design documentation?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the flight fare prediction code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

1 Architecture



2 Architecture design

This project is to create an interface for the user to know their approximate flight ticket price, in addition to this, in need of getting the real time project experience we are importing the gathered data into our own database and then start the project from the scratch.

2.1 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>

2.2 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.

2.3 Upload data into Cassandra

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.

2.4 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.

2.5 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.

2.6 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.

2.7 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. Note I did not make the CSS and HTML File .

2.8 Data from user

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

2.9 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.

2.10 Rendering the results

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

2.11 Deployment

The tested model is then deployed to AWS EC2 Instance and local machine. So, users can access the project from any internet devices.

