# Architecture Design
# Flight Fare Prediction

Revision Number – 1.1

Last Date of Revision: 22 – 11 - 2022

Purna Jain

Document Version Control

| Date | Version | Description | Author |
|---|---|---|---|
| 11 – 11 - 2022 | 1.0 | Abstract<br>Introduction<br>Architecture | Purna |
| 18 – 07 - 2022 | 1.1 | Architecture Design | Purna |

# Contents

# Abstract

The recent changes in the international market had a large impact on the Aviation sector because of several reasons. These impact the two class folks, the first is Business perspective and second is Customer perspective. The major reason for such an impact is the governments around the world amended totally different rules to their various Airline firms. Taking these factors into consideration, the value of the flight tickets has varied from one place to another. Booking a flight ticket has its price tag split into two, one is online bookings and other is offline bookings. Each of these have their various criteria for value of the price, one such example is the server load and therefore the range of booking requests. During this machine learning implementation, we are going to see numerous factors that impact the price of the flight ticket and predict the acceptable price of the ticket.

# Introduction

## 1.1 Why this Architecture Design Document ?

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.

## 2. Architecture
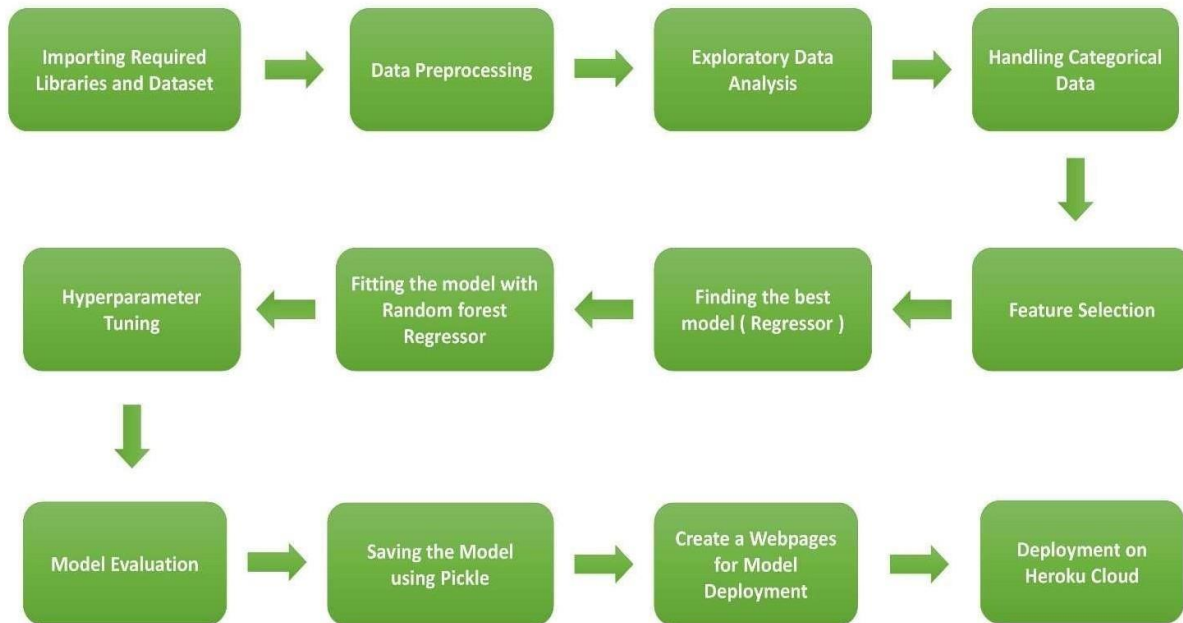


## 3. Architecture Design

### 3.1 Data Collection

The data for these 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)

### 3.2 Data Description

Flight Fare Prediction is 10K+ dataset publicly available on Kaggle. The information in the dataset is present in two separate excel files named as train.xlsx and test.xlsx. Dataset contains 10683 rows which shows the information such Date of Journey, Source, Destination, Arrival Time, Departure Time, Total stops, Airlines, Additional Info and Price. The glance of the Dataset is :

| Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info | Price |
|---|---|---|---|---|---|---|---|---|---|---|
| IndiGo | 24/03/2019 | Banglore | New Delhi | BLR → DEL | 22:20 | 01:10 22 Mar | 2h 50m | non-stop | No info | 3897 |
| Air India | 1/05/2019 | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 13:15 | 7h 25m | 2 stops | No info | 7662 |
| Jet Airways | 9/06/2019 | Delhi | Cochin | DEL → LKO → BOM → COK | 09:25 | 04:25 10 Jun | 19h | 2 stops | No info | 13882 |
| IndiGo | 12/05/2019 | Kolkata | Banglore | CCU → NAG → BLR | 18:05 | 23:30 | 5h 25m | 1 stop | No info | 6218 |
| IndiGo | 01/03/2019 | Banglore | New Delhi | BLR → NAG → DEL | 16:50 | 21:35 | 4h 45m | 1 stop | No info | 13302 |
| SpiceJet | 24/06/2019 | Kolkata | Banglore | CCU → BLR | 09:00 | 11:25 | 2h 25m | non-stop | No info | 3873 |
| Jet Airways | 12/03/2019 | Banglore | New Delhi | BLR → BOM → DEL | 18:55 | 10:25 13 Mar | 15h 30m | 1 stop | In-flight meal not included | 11087 |
| Jet Airways | 01/03/2019 | Banglore | New Delhi | BLR → BOM → DEL | 08:00 | 05:05 02 Mar | 21h 5m | 1 stop | No info | 22270 |
| Jet Airways | 12/03/2019 | Banglore | New Delhi | BLR → BOM → DEL | 08:55 | 10:25 13 Mar | 25h 30m | 1 stop | In-flight meal not included | 11087 |
| Multiple carriers | 27/05/2019 | Delhi | Cochin | DEL → BOM → COK | 11:25 | 19:15 | 7h 50m | 1 stop | No info | 8625 |
| Air India | 1/06/2019 | Delhi | Cochin | DEL → BLR → COK | 09:45 | 23:00 | 13h 15m | 1 stop | No info | 8907 |
| IndiGo | 18/04/2019 | Kolkata | Banglore | CCU → BLR | 20:20 | 22:55 | 2h 35m | non-stop | No info | 4174 |
| Air India | 24/06/2019 | Chennai | Kolkata | MAA → CCU | 11:40 | 13:55 | 2h 15m | non-stop | No info | 4667 |
| Jet Airways | 9/05/2019 | Kolkata | Banglore | CCU → BOM → BLR | 21:10 | 09:20 10 May | 12h 10m | 1 stop | In-flight meal not included | 9663 |

## 3.3 Importing data into Database

Created associate API for the transfer of the info into the Cassandra info, steps performed are :

- Connection is created with the info.
- Created a info with name FlightInfo.
- cqlsh command is written for making the info table with needed parameters.
- And finally, a cqlsh command is written for uploading the Knowledge Set into data table by bulk insertion.

### 3.4 Exporting Data from Database

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

### 3.5 Data Preprocessing

- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Converted all the desired columns into Datetime format.

- Performed One – Hot encoding on the desired columns.
- Checking the distribution of the columns to interpret its importance.

Now, the info is prepared to train a Machine Learning Model.

### 3.6    Modeling Process

After preprocessing the data, We visualize our data to gain insights and then these insights are randomly spread and split into two parts, train and test data. After splitting the data, we use Random Forest Regressor to model our data to predict the Flight Fare price.
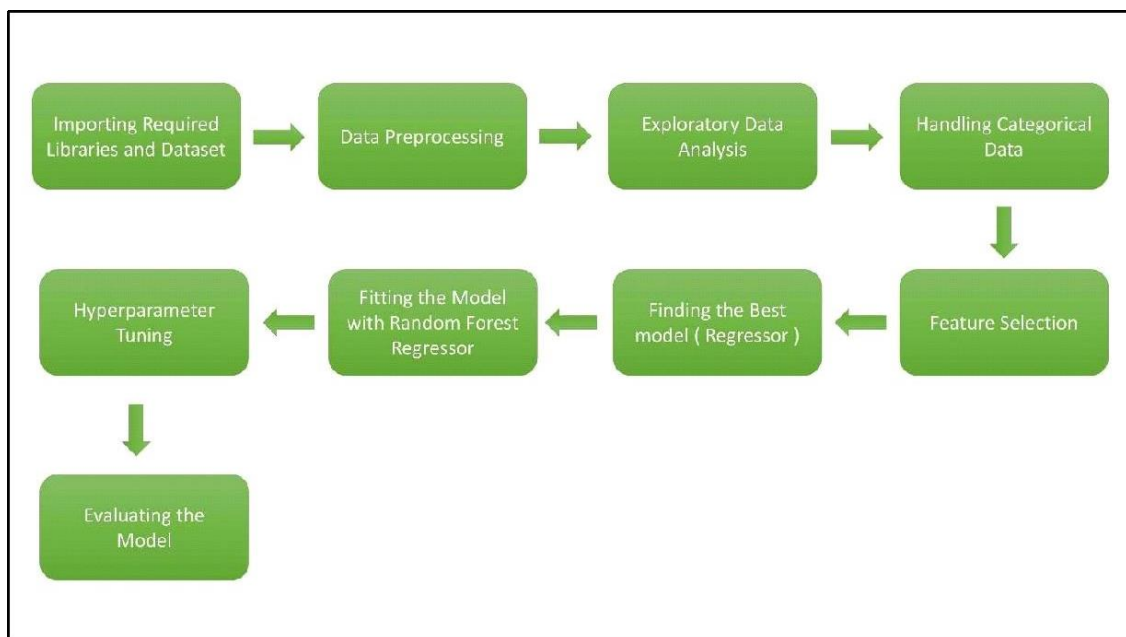


### 3.7        UI Integration

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.

3.8        Data from User

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

3.9        Data Validation

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

3.10        Rendering the Results

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

3.11        Deployment

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