Architecture Design
Flight Fare Estimator

Revision Number – 1.1
Last Date of Revision: 22–01–2023

Shubham Shaah

Document Version Control

Date	Version	Description	Author
10 – 11 - 2022	1.0	Abstract Introduction Architecture	Shubham
22 – 01 – 2023	1.1	Architecture Design	Shubham

Contents

Abstract

The aviation market has seen significant changes and challenges in recent years. The COVID-19 pandemic has had a major impact on the industry, causing widespread disruptions and reductions in demand for air travel. Despite this, the industry is beginning to recover as vaccine rollouts and declining case numbers lead to increased consumer confidence. However, the industry is still facing challenges, such as shifting consumer preferences for travel and a rise in low-cost carriers. In response, many airlines and travel companies are investing in new technologies, such as flight fare prediction systems, to remain competitive and meet the evolving needs of travellers.

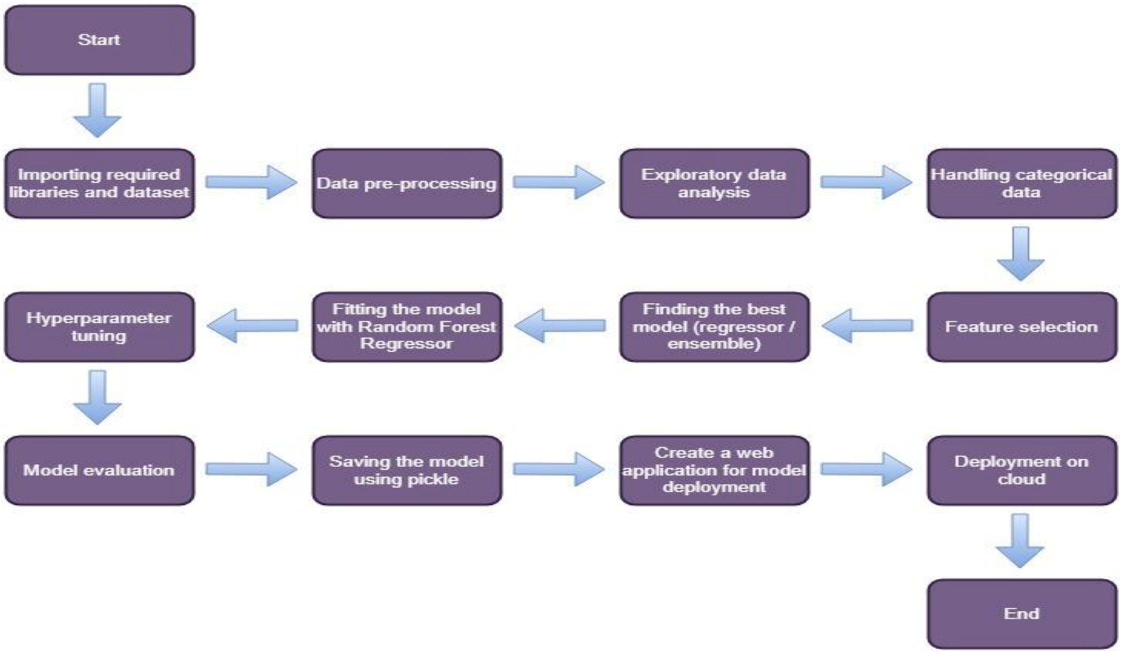
The prediction of flight fares is important because it helps travellers to make informed decisions when booking their travels. By having access to accurate and up-to-date predictions of flight fares, travellers can save money and avoid overpaying for their flights. Additionally, a prediction system for flight fares can also help airlines and travel companies to optimize pricing strategies and make data-driven decisions. In a highly competitive market, being able to predict flight fares can provide a significant advantage to both travellers and companies.

1. Introduction

1.1. Why 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


Start
Importing required libraries and dataset
Data pre-processing
Exploratory data analysis
Handling categorical data
Feature selection
Finding the best model (regressor / ensemble)
Fitting the model with Random Forest Regressor
Hyperparameter tuning
Model evaluation
Saving the model using pickle
Create a web application for model deployment
Deployment on cloud
End

3. Architecture Design

3.1. 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

3.2. Data Description

Flight Fare Prediction is 10K+ dataset publicly available on Kaggle. The information in the dataset is present in 2separate 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:

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 info with name Flight Info.
- The command is written for making the info table with needed parameters.
- And finally, a 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.

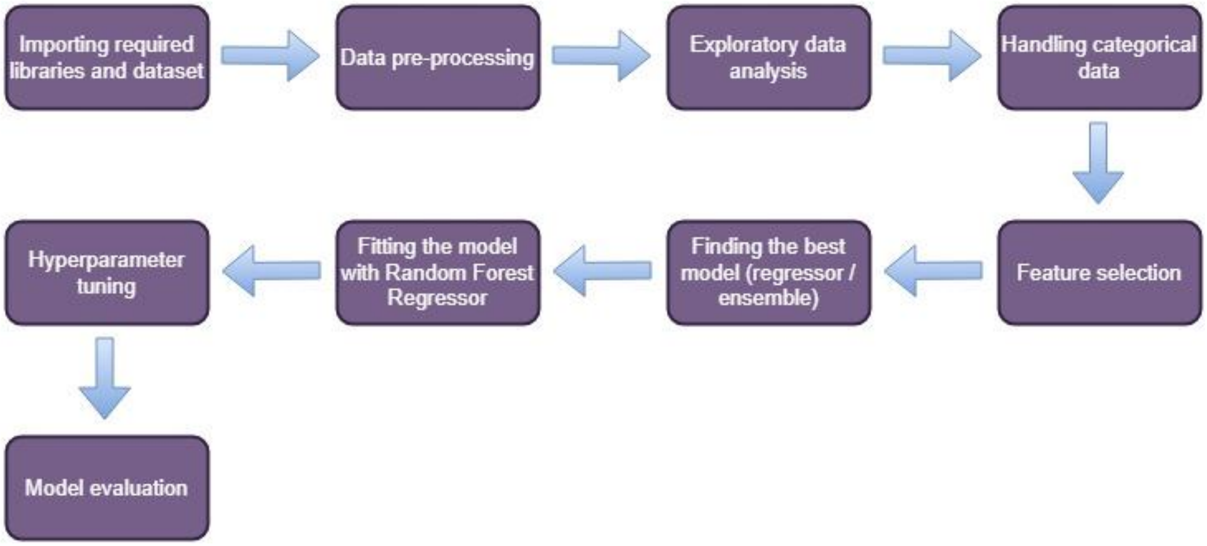
3.5. Data Pre-processing

- 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. Modelling Process

After pre-processing 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.


Importing required libraries and dataset
Data pre-processing
Exploratory data analysis
Handling categorical data
Feature selection
Finding the best model (regressor / ensemble)
Fitting the model with Random Forest Regressor
Hyperparameter tuning
Model evaluation

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 result

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

3.11. Deployment

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