Architecture Design
Flight Fare Prediction

Revision Number – 1.1

Last Date of Revision: 24 – 04 - 2023

ANKIT

MUDIT ARORA

M.S.S.S.CHAITANYA

Document Version Control

Date	Version	Description	Author
19 – APRIL - 2023	1.0	Abstract Introduction Architecture	M.S.S.S.CHAITANYA, ANKIT
20 – APRIL - 2023	1.1	Architecture Design	MUDIT ARORA

Contents

Abstract

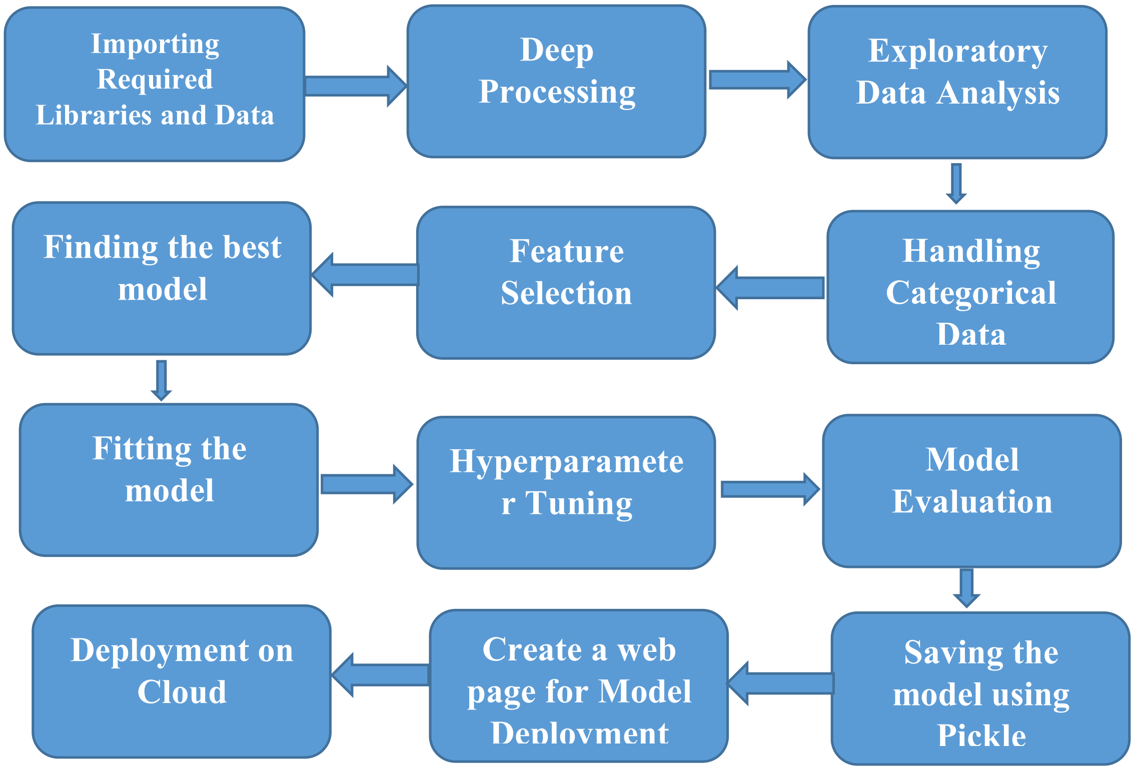
The Aviation sector faced significant challenges due to the recent changes in the international market. These changes affected both the Business and Customer perspectives of the industry. One of the main reasons for this was the different regulations imposed by governments around the world on their Airline companies. As a result, the flight ticket prices varied widely across different locations. There were also two modes of booking a flight ticket: online and offline. Each mode had its own factors that influenced the price, such as the server load and the number of booking requests. In this machine learning implementation, we will explore various factors that affect the flight ticket price and predict the optimal price for the ticket.

1. 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


Importing Required Libraries and Data
Deep Processing
Exploratory Data Analysis
Handling Categorical Data
Feature Selection
Finding the best model
Fitting the model
Hyperparamete r Tuning
Model Evaluation
Saving the model using Pickle
Create a web page for Model Deployment
Deployment on Cloud

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

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 :

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.

4.Deployment

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