Architecture Design - Flight Fare Prediction

Revision Number – 1.1
Last Date of Revision: 01/08/2023

Author : Aditya Singh Jadon

Document Version Control

Date Version Description Author : 30/08/2023
1.0
Abstract
Introduction
Architecture
30/07/2023

1.1
Architecture Design

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.

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 Design

2.1 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

2.2 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 as Date of Journey, Source, Destination, Arrival Time, Departure Time, Total stops, Airlines, Additional Info and Price.

2.3 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 an info with the name FlightInfo.
•cqlsh command is written for making the info table with needed parameters.
•And finally, a command is written for uploading the Knowledge Set into the data table by bulk insertion.

•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.

2.4 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.
• Convert 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.

•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.

•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.

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

•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.

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

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