ARCHITECTURE DESIGN

Flight Fare Prediction

HARSHA MS

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 31/10/2022 | Harsha MS | Architecture |
|  |  |  |  |

Document Version control----------------------------------------------------------2

1 Why this Architecture Design Document? -----------------------------------------4

2 Architecture------------------------------------------------------------------------- 5

3 Architecture Design------------------------------------------------------5

Data Collection-----------------------------------------------------------5

Data Description-----------------------------------------------------------5

Data Clustering-------------------------------------------------------------5

Importing Data into Database---------------------------------------------5

Exporting Data from Database -------------------------------------------5

Data Pre-processing---------------------------------------------------------6

Modelling Process-----------------------------------------------------------6

UI Integration----------------------------------------------------------------6

Data from User-------------------------------------------------------------6

Data Validation------------------------------------------------------------6

Rendering the results------------------------------------------------------6

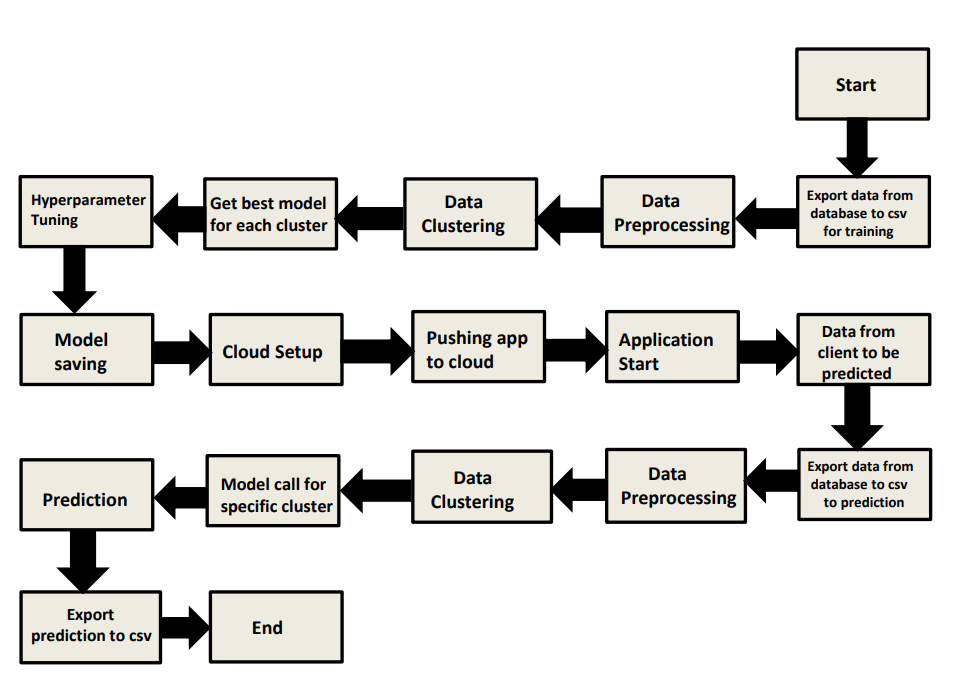
4 Deployment------------------------------------------------------------------------6

**INTRODUCTION**

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.

1. **Architecture**

****

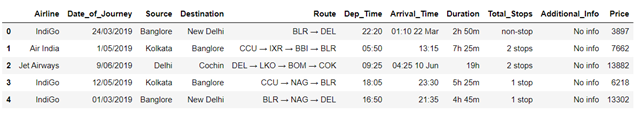
1. **Architecture Design**

**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

**Data Description**

Flight Fare Prediction is a 10K+ dataset publicly available on the Kaggle. The information in the dataset is present in two separate excel files named as train.xlsx and test.xlsx. The 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. A glance of the Dataset is:



**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 Flight Info.

•cqlsh command is written for making the info table with needed

•And finally, a cqlsh command is written for uploading the Knowledge Set into 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.

**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.

**Model 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.

**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.

**4 Deployment**

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