FLIGHT FARE PREDICTION LLD

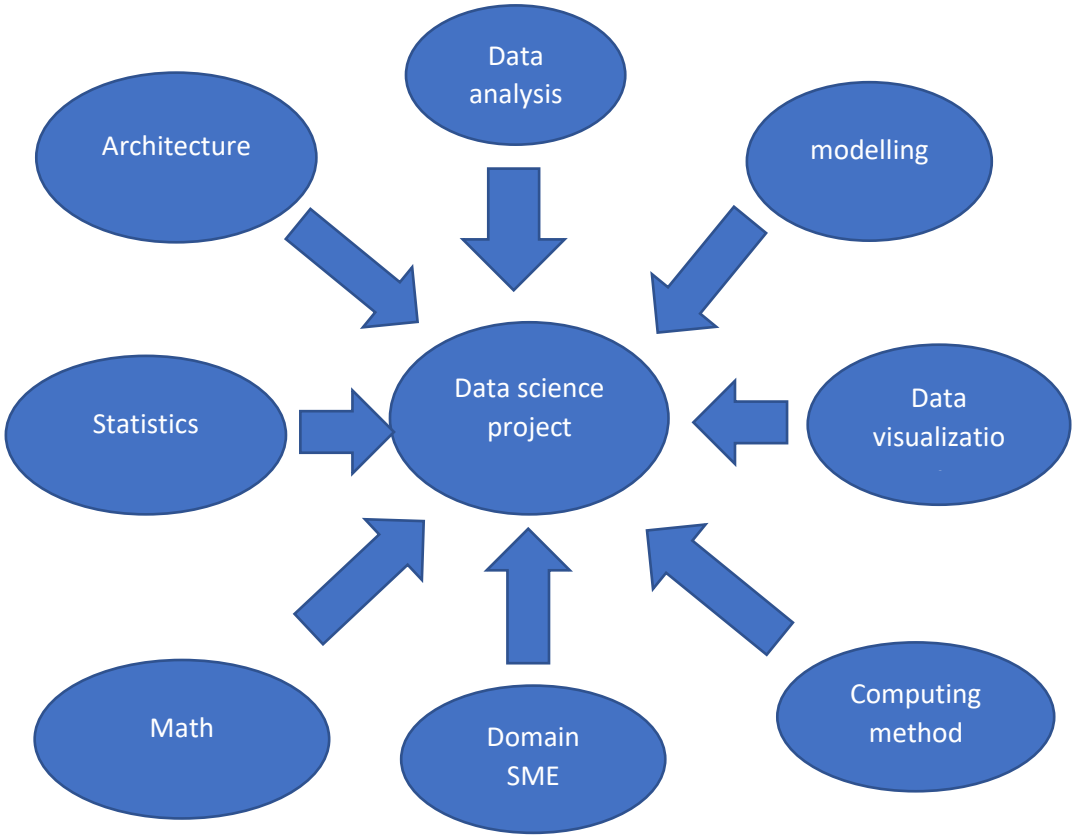
Contents

Abstract

Travelling through flights has become a new normal for most people in this contemporary scenario. Flight fares are affected by so many factors like place, time, stoppages, Destination, Arrival time, route and so on. With the break through in AI and ML we are able to predict the flight fares using many algorithms. In this project we use, regression methodology to come to the conclusion of flight fare which consists basically of two data sets mainly training data and testing data.

Introduction

1.1 Architecture



2. Architecture Design

In this project, we have implemented the machine learning algorithm to create a basic web application which will predict the flight prices by applying machine learning algorithms to historical flight data.

2.1 Data Gathering

The data for this project is divided into two i.e, one being training data and another is test dataset. Both are collected from the Kaggle website.

2.2 Tool Used

- Python 3.10

- frameworks like numpy, pandas, sci-kit learn and alternative modules for building the model.

- Visual Studio code IDE. For visualizations seaborn and components of matplotlib are used.

2.3 Data Description

We basically have the dataset divided into two files being train data and test data where there are 11 features totally. 10 being input features and price feature being output feature. This data set consists of 10k records approximately.

2.4 Import 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 name flight fare.

- Cqlsh command is written for making the info table with needed parameters.

- And finally, a “cqlsh” command is written for uploading the knowledge set data into the table by bulk insertion.

2.5 Export Data into Database

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

2.6 Data Pre processing

Steps performed in pre-processing are:

● Check for null values.

● Convert the string data type into the desired data type.

● Extract the Date Column into three separate column of Day, Month, Year likewise.

●Extract the hour and minute from Departure time(cut off the seconds).

● Perform Label Encoding and hot-Encoding to convert Categorical values to model Identifiable values.

● Build the machine learning algorithm using processed data and save the model in “.pkl” format.

2.7 Modelling

After feature engineering, we get into the stage of model building where we use multiple Machine learning algorithms to come to a conclusion which gives us a better accuracy and precision. In our case, we got a better accuracy using Random forest Regressor. Below are the algorithms we’ve used.

- Linear Regression
- Random Forest Regressor
- XG Boost Regressor

- Gradient Boosting Regressor

2.8 UI Integration

UI integration has been done with the classic front end tools like html, CSS and bootstrap. But in order to get the data from backend we use Flask to develop API. When user enter the input data the data will be processed in the back end and will be visible on the web page.

2.9 Input from user

The input is taken from user on the web page that's visible on the browser.

2.10 Data Validation

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

3. Deployment

The web app is being deployed on render.com which a cloud hosting platform for web apps developed under various dynamic languages.

3.1 Unit test

Data	**Anticipated result**	**Obtained result**	**Status**
UI	Verify if all the features of input is taking values.	yes	Passed
UI	Verify whether system prompts to fill all fields if not filled	yes	passed
UI URL	Verify whether host URL is accessible	yes	Passed
UI URL	Verify whether the UI url loads completely or not	yes	Passed