Name: Mohini Kalbandhe

Batch code:LISUM10

Submission Date: 29th June 2020

Submitted to : Data Glacier Team

Flight Price Prediction

Table of Content

About flight fare prediction project

In this Project, we will predict air ticket prices using information like airline, date, locations. We will use random forest tree as it offers both decent speed and accuracy.

The basic idea behind the algorithm is to **find the point** in the independent variable to **split the data-set** into 2 parts, so that the **mean squared error** is the minimized at that point. The algorithm does this in a repetitive fashion and forms a tree-like structure.

Input:

- Airline
- Date
- Destination and origin
- .....

Source:

- https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh

Output:

- Ticket Price

Techniques:

- Feature Engineering
  - Categorical Input
  - Seasonality
  - Time of day
- Feature Selection
  - Heat Map
  - Extra Trees Regressor¶
- Random Forest Tree Regression
- Random Search Hyperparameter Tuning

Overview:

Importing dataset

1. Since data is in form of excel file we have to use pandas read_excel to load the data
2. After loading it is important to check the complete information of data as it can indication many of the hidden infomation such as null values in a column or a row
3. Check whether any null values are there or not. if it is present then following can be done,
   A. Imputing data using Imputation method in sklearn
   B. Filling NaN values with mean, median and mode using fillna() method
4. Describe data --> which can give statistical analysis

EDA

From description we can see that Date_of_Journey is a object data type,\ Therefore, we have to convert this datatype into timestamp so as to use this column properly for prediction

For this we require pandas **to_datetime** to convert object data type to datetime dtype.

Handling Categorical Data

One can find many ways to handle categorical data. Some of them categorical data are,

1. **Nominal data** --> data are not in any order --> **OneHotEncoder** is used in this case
2. **Ordinal data** --> data are in order --> **LabelEncoder** is used in this case

Feature Selection

Finding out the best feature which will contribute and have good relation with target variable. Following are some of the feature selection methods,

1. **heatmap**
2. **feature_importance_**
3. **SelectKBest**

Fitting model using Random Forest

1. Split dataset into train and test set in order to prediction w.r.t X_test
2. If needed do scaling of data
    - Scaling is not done in Random forest
3. Import model
4. Fit the data
5. Predict w.r.t X_test
6. In regression check **RSME** Score
7. Plot graph

Hyperparameter Tuning

- Choose following method for hyperparameter tuning
    1. **RandomizedSearchCV** --> Fast
    2. **GridSearchCV**
- Assign hyperparameters in form of dictionery
- Fit the model
- Check best paramters and best score

Save the model to reuse it again

Model saved in model.pkl file, so that we can use it to run the app.

The model's performs much better than the baseline model.

Baseline Model

- MAE: 1196
- MSE: 4519902
- RMSE: 2126

Best Model

- MAE: 1170

- MSE: 4053505
- RMSE: 2013

Creating app.py:

Application is created using flask. This code we have so far builds simple website.

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
return "Hey there!"

if __name__ == '__main__':
app.run(debug=True)
```

The pickle file which we have created and saved “flight_rf.pkl”

Model = pickle.load(open(“flight_rf.pkl”, “rb”) using this command we can read pickle file inside the app.

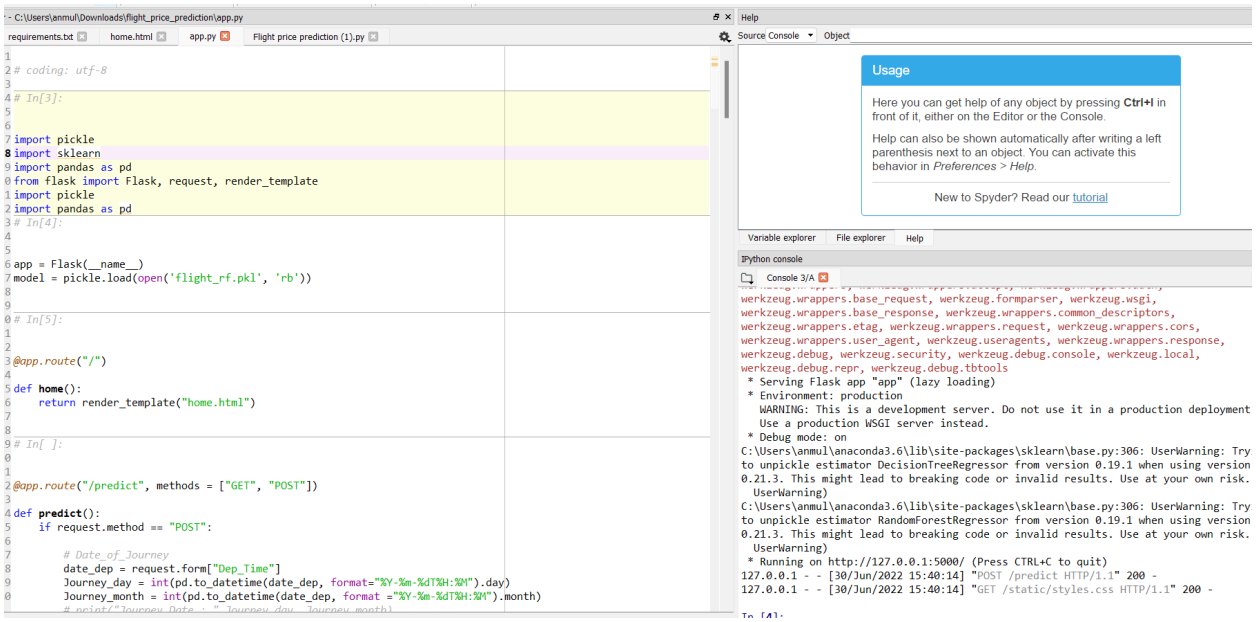
Rendor_template is used to read html file”home.html”

With the help of this command “@app.route("/predict", methods = ["GET", "POST"])” created all the tabs and helps to access

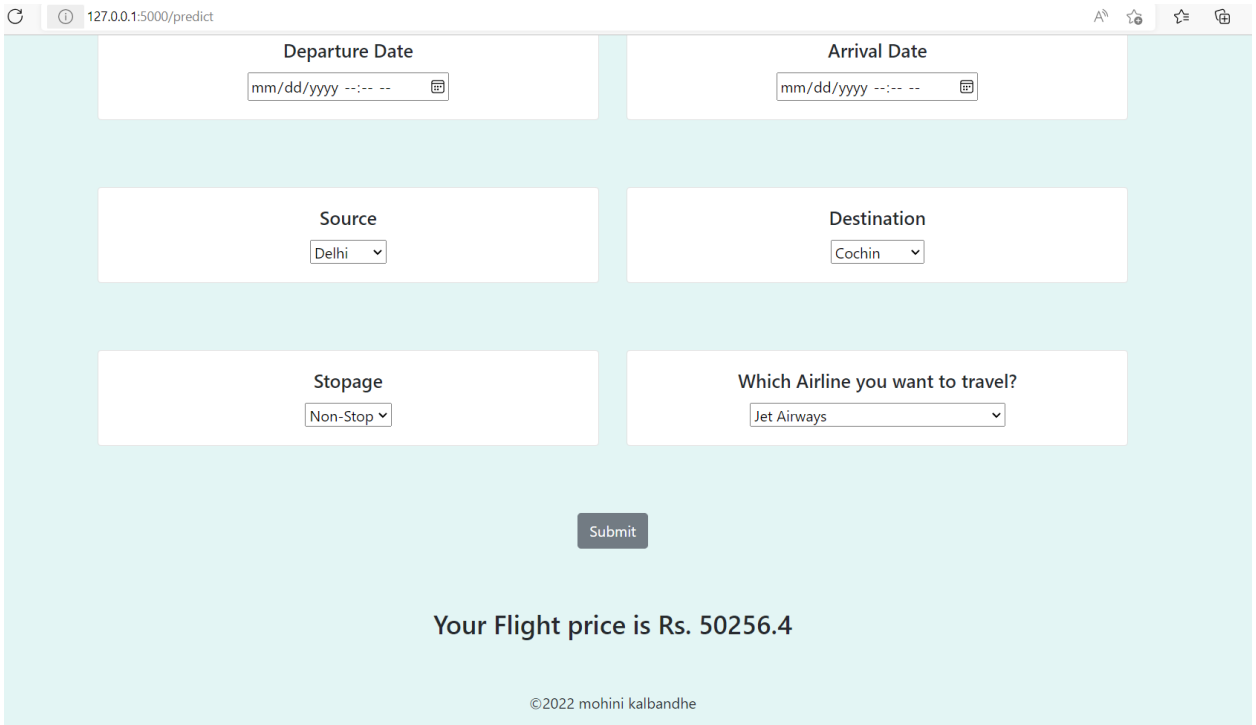
All the body of tabs is created in html file that will be available to access below this.

Created another file named “style.css” used to provide color and other features, under template folder there is a file called “home.html”, used in app.py

Tested file on Spyder ipython consol, and got the address, as shown below



As soon as I ran the address http://127.0.0.1:5000/ I got the following result, I entered the date and time and fill up all necessary details, clicked on submit button, the app got predict the results as fallows.



Installation:

Procfile is important to access web application named gunicorn and app:app shows app name which we have provided in app.py and app.py itself.

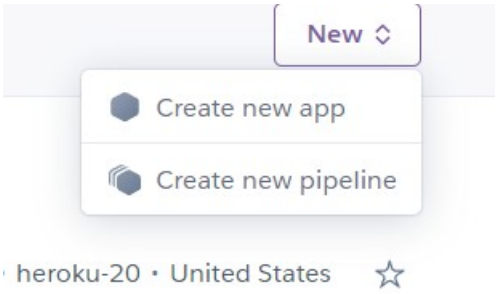
Requirements.txt file is created using command “pip install –r requirements.txt” , which is useful to download the requirements in heroku app.

These are all important files to create a web app, Let’s move towords Deployment Phase.

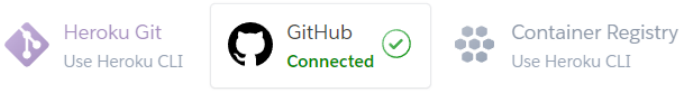
Deployment using Heroku:

First of all we need to create an account in heroku app and login in that app. You can see the name of display, the click on new tab and create new app.

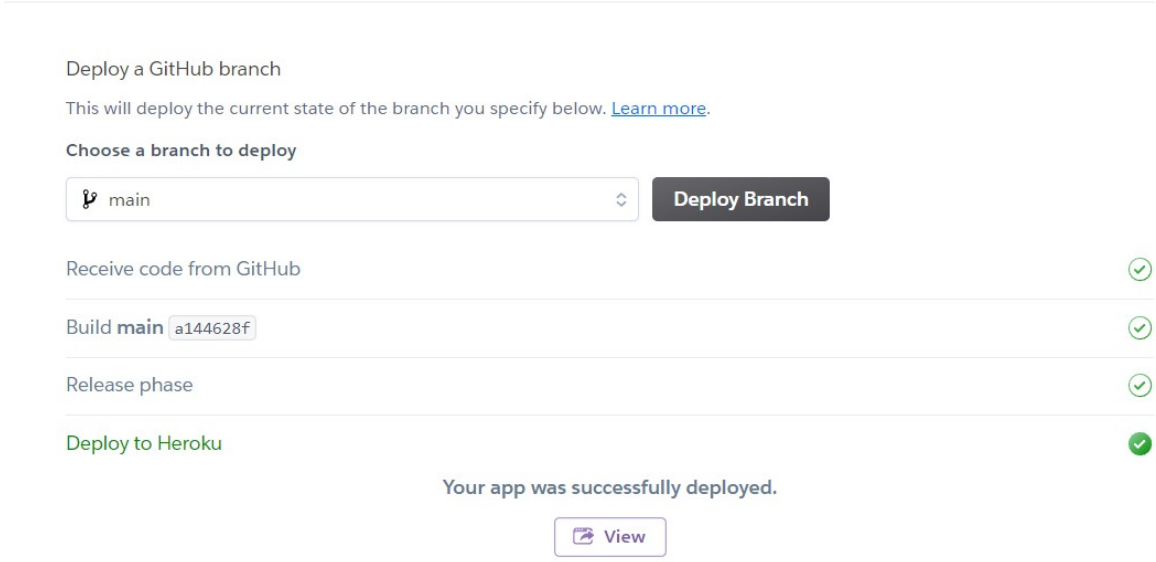
You can see the name of app in upper left corner.



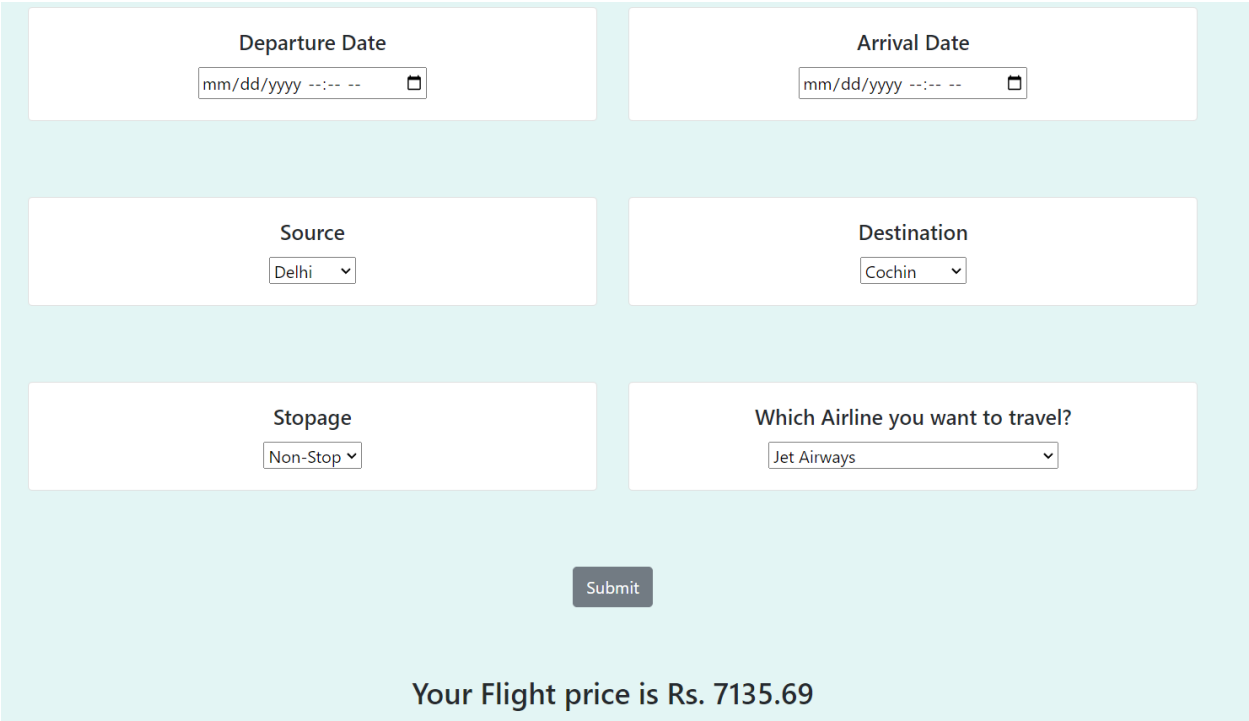
Then there is an option to select weather you want use heroku cli manually or you want to connect file to GitHub , I did it on github by connecting my project folder with heroku app. The next step is to follow Heroku documentation.



After connecting the project we can deploy it manually or automatically, I did manual deployment ,by clicking on deploy branch all the files in requirement.txt will get download to create an app. Finally we can see the command “your app has been successful installed, there is a tab “view”, just click on it and you can see the productionized app on website.



You can see my app on the following website https://flight-fare-prediction-099.herokuapp.com/Predict. App displays as follows



The next step is to select date time and all the necessary option click on “submit” button you can see the Price prediction results.

Directory Tree:

```
├── static
│   ├── css
├── template
│   ├── home.html
├── Procfile
├── README.md
├── app.py
├── flight_price.ipynb
├── flight_rf.pkl
├── requirements.txt
```

Technologies Used:

Reference :

Data is used for kaggle dataset

“https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh”