

**Name: Mohini Kalbandhe**

**Batch code: LISUM10**

**Submission Date: 29<sup>th</sup> June 2020**

**Submitted to : Data Glacier Team**



# Flight Price Prediction

## Table of Content

- About flight fare prediction project
- Overview
- Creating app.py
- Installation
- Deployment on Heroku
- Directory Tree
- Technologies used
- References

## 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 indicate many of the hidden information 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 dictionary
- 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

## 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.

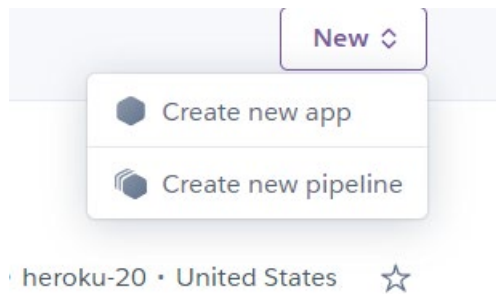
Requirements.txt file is created using command “pip install -r requirements.txt” , which is useful to download the requirements in heroku app.

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

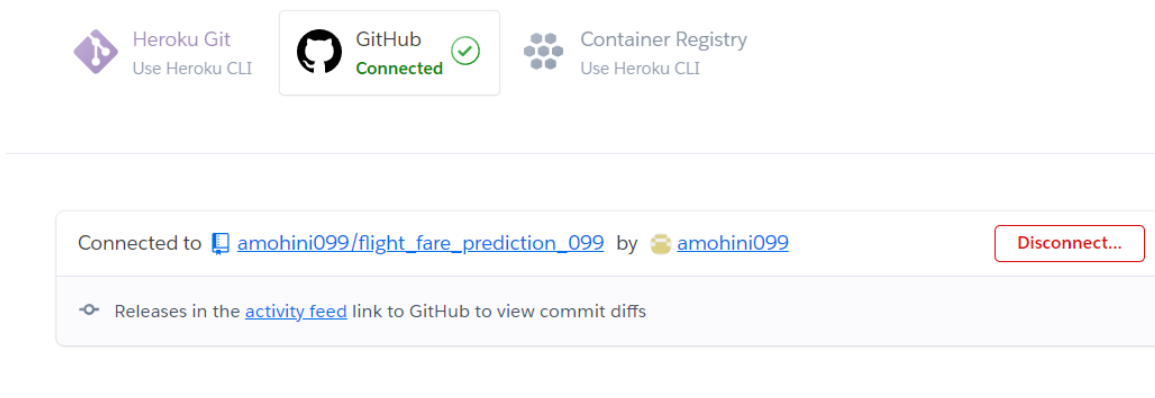
## Deployment using Heroku:

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 whether you want to use Heroku CLI manually or you want to connect 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 'requirements.txt' will get downloaded to create an app. Finally we can see the command "your app has been successfully installed, there is a tab 'view', just click on it and you can see the productionized app on website."

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

 main

Deploy Branch

Receive code from GitHub 

Build main `a144628f` 

Release phase 

Deploy to Heroku 

Your app was successfully deployed.

 View

You can see my app on the following website <https://flight-fare-prediction-099.herokuapp.com/Predict>.

App displays as follows

Departure Date

Arrival Date

Source

Destination

Stopage

Which Airline you want to travel?

Submit

Your Flight price is Rs. 7135.69

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"](https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh)