FLIP ROBO

FLIGHT PRICE PREDICTION

Submitted by:

MITESH VERMA

ACKNOWLEDGMENT

I would like to acknowledge Mr. Shwetank Mishra (FlipRobo) for his vital cooperation and help in ensuring the successful completion of my assignment. He deserves the utmost credit for the assignment’s outcome.

Finally, I would want to convey my sincere thanks Datatrained Academy and their guidance without them, the task would not have been accomplished.

The website that I referred are:

https://learning.datatrained.com

https://www.w3schools.com

https://www.freecodecamp.org

https://github.com

https://www.geeksforgeeks.org

https://www.yatra.com

https://www.ixigo.com

https://stackoverflow.com

https://www.kaggle.com

INTRODUCTION

- Business Problem Framing

  Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on -

  1. Time of purchase patterns (making sure last-minute purchases are expensive)
  2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases).

- Conceptual Background of the Domain Problem

  This project is about predicting the price of Flights in India, using the data of some websites. There are some phases in this project:

  → Data Collection Phase.
  → Data Analysis.
  → Model Building Phase.

- Review of Literature

  1. First phase is Data Scraping using Selenium.

  Scraped data from:

  – www.yatra.com
  – www.ixigo.com

Features:

- Flight: The name of the airline.
- Date: The date of the journey
- Duration: Total duration of the flight.
- Source: The source from which the service begins.
- Destination: The destination where the service ends.
- Departure: The time when the journey starts from the source.
- Arrival: Time of arrival at the destination.
- Stops: Total stops between the source and destination.
- Meal: Meal will be there during the journey or not.

Target:

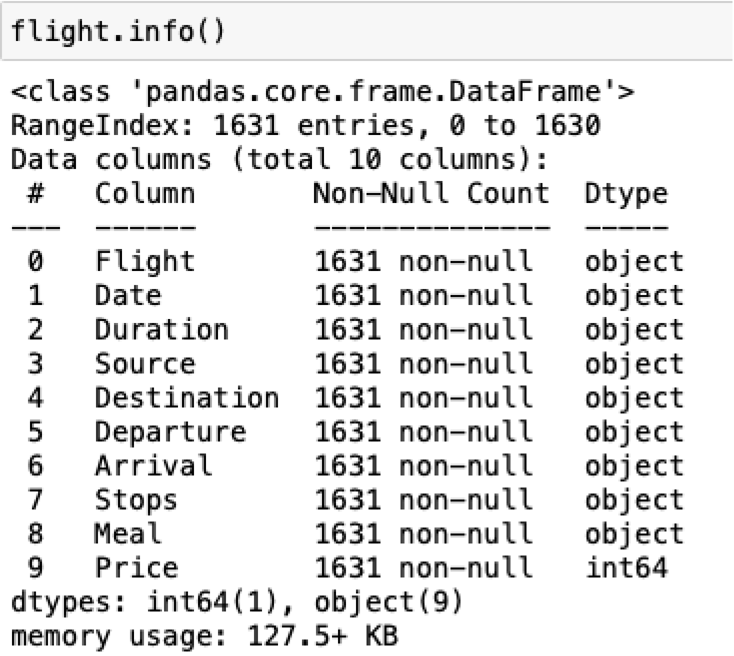
- Price: The price of the ticket

Motivation for the Problem Undertaken

This project is on the data scraping, data science and machine learning model, build the model to predict the flight price based on some features.

Analytical Problem Framing

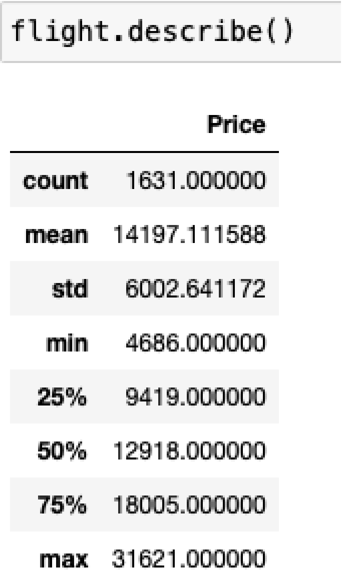
- Mathematical/ Analytical Modeling of the Problem
  - **Information of the dataset:**



```
flight.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1631 entries, 0 to 1630
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Flight       1631 non-null   object
 1   Date         1631 non-null   object
 2   Duration     1631 non-null   object
 3   Source       1631 non-null   object
 4   Destination  1631 non-null   object
 5   Departure    1631 non-null   object
 6   Arrival      1631 non-null   object
 7   Stops        1631 non-null   object
 8   Meal         1631 non-null   object
 9   Price        1631 non-null   int64 
dtypes: int64(1), object(9)
memory usage: 127.5+ KB
```

  - **Description of the dataset:**



```
flight.describe()
```

	Price
count	1631.000000
mean	14197.111588
std	6002.641172
min	4686.000000
25%	9419.000000
50%	12918.000000
75%	18005.000000
max	31621.000000

- Data Sources and their formats

  → Data Collection Phase.

  i. Collected the data from different websites such as www.yatra.com, www.ixigo.com .
  ii. Collected data like Flight, Date, Duration, Source, Destination, Departure, Arrival, Stops, Meal.
  iii. Saved the dataset as a csv file.
  iv. Data cleaning from excel and through python.

  → Model Building Phase.

  i. Data Cleaning.
  ii. EDA
  iii. Visalization
  iv. Data Pre-processing
  v. Model Building
  vi. Selecting the best model
  vii. Hyperparameter tuning

- Data Pre-processing Done
  - **EDA**
  - **Description**
  - **No null present**
  - **Data cleaning**
  - **Visualization**
  - **Encoding**

- Hardware and Software Requirements and Tools Used

Anaconda-navigator

jupyter notebook

matplotlib-inline==0.1.6

numpy==1.23.2

packaging==21.3

pickleshare==0.7.5

platformdirs==2.5.2

prompt-toolkit==3.0.30

pyparsing==3.0.9

python-dateutil==2.8.2

scikit-learn==1.1.2

scipy==1.9.0

sklearn==0.05

Model/s Development and Evaluation

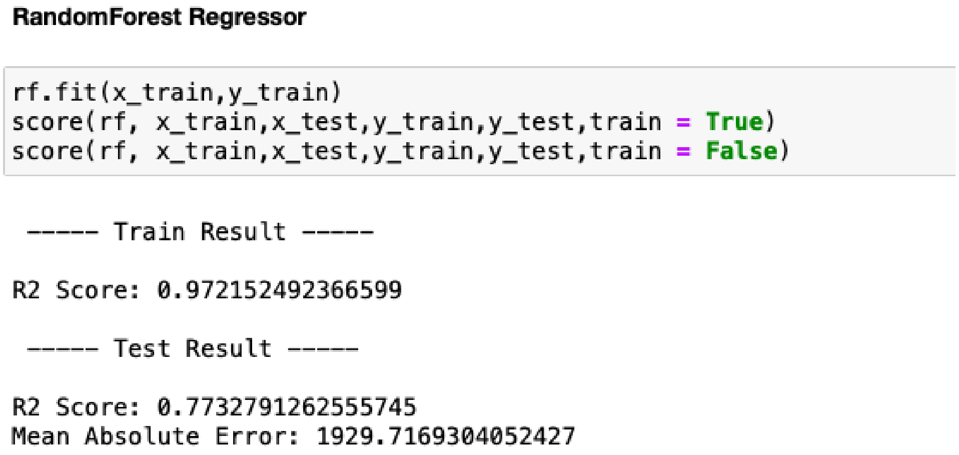
- Identification of possible problem-solving approaches (methods)
  - **EDA**
  - **Description**
  - **No null present**
  - **Data cleaning**
  - **Visualization**
  - **Encoding**
  - **Model Building**
  - **Select the best model**
  - **Hyperparameter tuning**

- Testing of Identified Approaches (Algorithms)

Algorithms used for the training and testing:

- **RandomForest Regressor.**
- AdaBoost Regressor.
- GradientBoosting Regressor.
- Super Vector Regressor.
- Kneighbors Regressor.

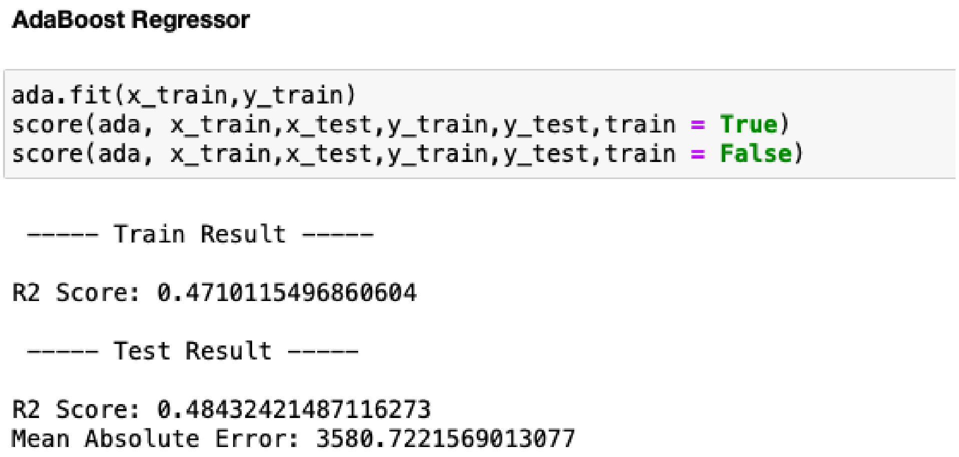
- Run and Evaluate selected models



RandomForest Regressor

```
rf.fit(x_train,y_train)
score(rf, x_train,x_test,y_train,y_test,train = True)
score(rf, x_train,x_test,y_train,y_test,train = False)
```

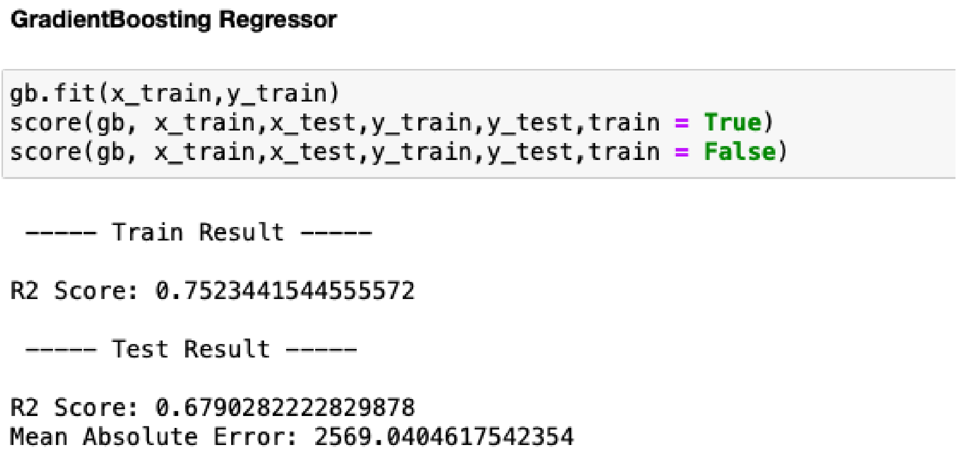
```
 ----- Train Result -----

R2 Score: 0.972152492366599

 ----- Test Result -----

R2 Score: 0.7732791262555745
Mean Absolute Error: 1929.7169304052427
```



AdaBoost Regressor

```
ada.fit(x_train,y_train)
score(ada, x_train,x_test,y_train,y_test,train = True)
score(ada, x_train,x_test,y_train,y_test,train = False)
```

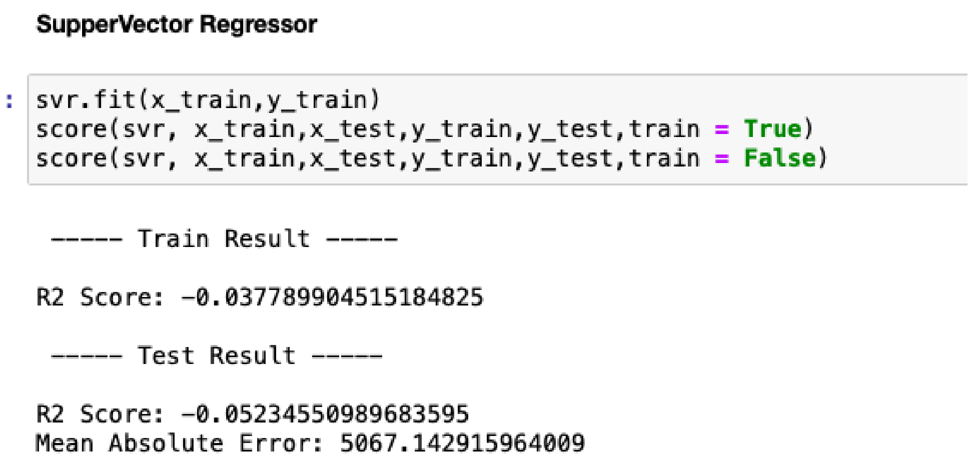
```
 ----- Train Result -----

R2 Score: 0.4710115496860604

 ----- Test Result -----

R2 Score: 0.48432421487116273
Mean Absolute Error: 3580.7221569013077
```



GradientBoosting Regressor

```
gb.fit(x_train,y_train)
score(gb, x_train,x_test,y_train,y_test,train = True)
score(gb, x_train,x_test,y_train,y_test,train = False)
```

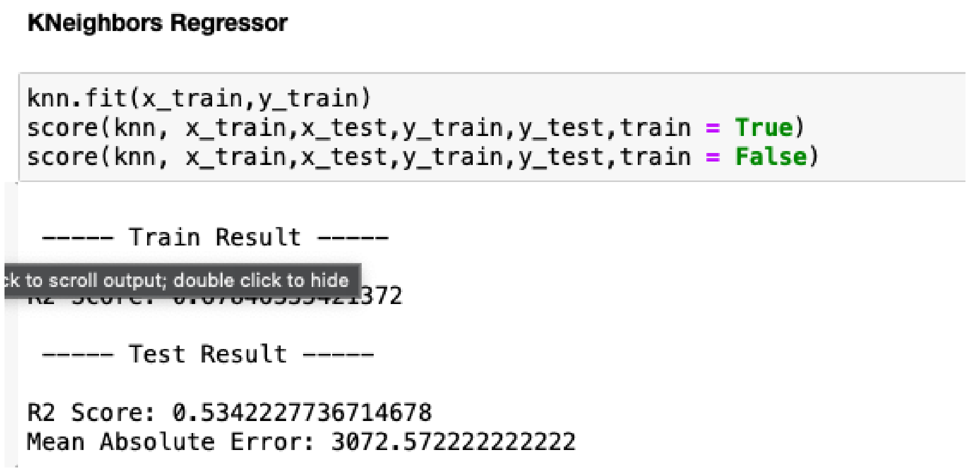
```
 ----- Train Result -----

R2 Score: 0.7523441544555572

 ----- Test Result -----

R2 Score: 0.6790282222829878
Mean Absolute Error: 2569.0404617542354
```



SupperVector Regressor

```
svr.fit(x_train,y_train)
score(svr, x_train,x_test,y_train,y_test,train = True)
score(svr, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

R2 Score: -0.037789904515184825

 ----- Test Result -----

R2 Score: -0.05234550989683595
Mean Absolute Error: 5067.142915964009
```



KNeighbors Regressor

```
knn.fit(x_train,y_train)
score(knn, x_train,x_test,y_train,y_test,train = True)
score(knn, x_train,x_test,y_train,y_test,train = False)
```

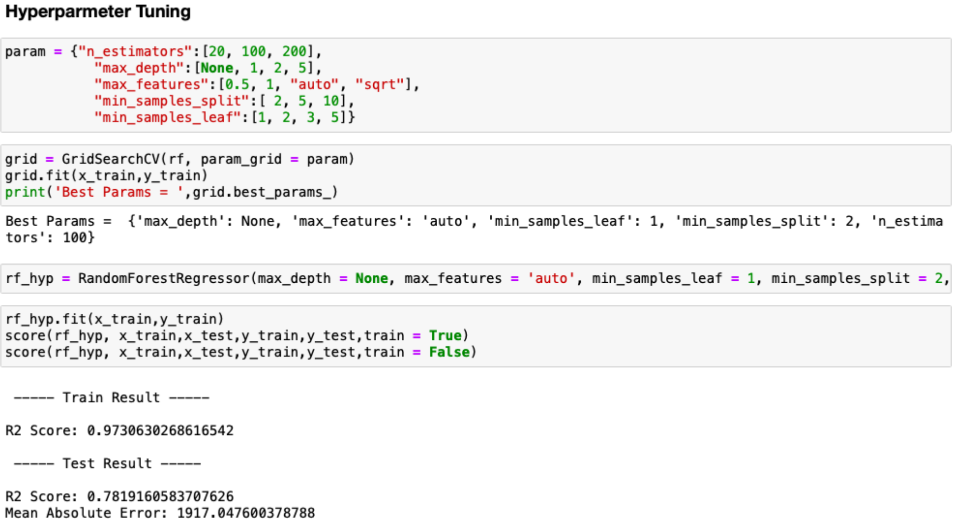
```
 ----- Train Result -----

R2 Score: [illegible]372

 ----- Test Result -----

R2 Score: 0.5342227736714678
Mean Absolute Error: 3072.572222222222
```

- Interpretation of the Results

RandomForest Regressor is giving the best score.

CONCLUSION

- Key Findings and Conclusions of the Study



Hyperparmeter Tuning

```
param = {"n_estimators":[20, 100, 200],
          "max_depth":[None, 1, 2, 5],
          "max_features":[0.5, 1, "auto", "sqrt"],
          "min_samples_split":[ 2, 5, 10],
          "min_samples_leaf":[1, 2, 3, 5]}
```

```
grid = GridSearchCV(rf, param_grid = param)
grid.fit(x_train,y_train)
print('Best Params = ',grid.best_params_)
```

```
Best Params =  {'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estima
tors': 100}
```

```
rf_hyp = RandomForestRegressor(max_depth = None, max_features = 'auto', min_samples_leaf = 1, min_samples_split = 2,
```

```
rf_hyp.fit(x_train,y_train)
score(rf_hyp, x_train,x_test,y_train,y_test,train = True)
score(rf_hyp, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

R2 Score: 0.9730630268616542

 ----- Test Result -----

R2 Score: 0.7819160583707626
Mean Absolute Error: 1917.047600378788
```

Post Tuning and Pre Tuning results are almost same for RandomForest Regressor.