FLIGHT PRICE PREDICTION

Submitted by:

SHAILAJ JOSHI

INTRODUCTION

- Problem Statement

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) So, you have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights. This project contains two phase-

Data Collection Phase
You have to scrape at least 1500 rows of data. You can scrape more data as well, it's up to you, More the data better the model

In this section you have to scrape the data of flights from different websites (yatra.com, skyscanner.com, official websites of airlines, etc). The number of columns for data doesn't have limit, it's up to you and your creativity. Generally, these columns are airline name, date of journey, source, destination, route, departure time, arrival time, duration, total stops and the target variable price. You can make changes to it, you can add or you can remove some columns, it completely depends on the website from which you are fetching the data.

Data Analysis Phase:
After cleaning the data, you have to do some analysis on the data. Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time? What is the best time to buy so that the consumer can save the most by taking the least risk? Does price increase as we get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

Model Building Phase:

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Follow the complete life cycle of data science. Include all the steps like

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

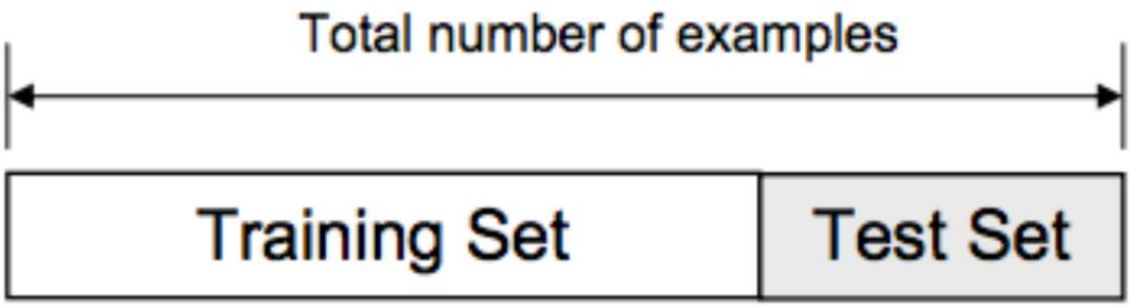
Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

  - Train/Test Split:

  The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset



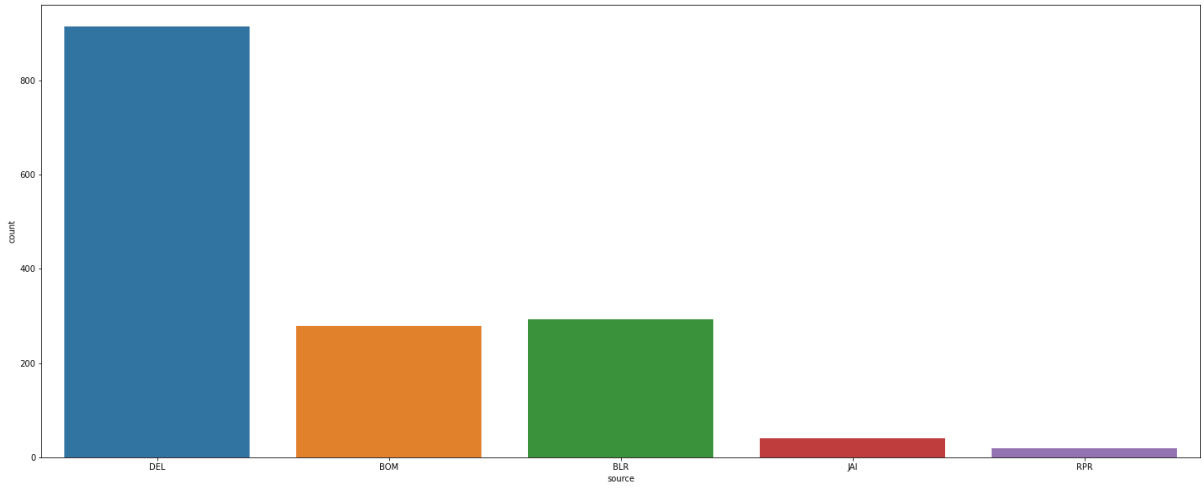
  Using the random state, we divide the training and testing data in the ratio of 0.8 and 0.2 meaning 20% of the data is the testing data and training data is 80%

- Data Sources and their formats

Scrapped 1500+ data from skyscanner website and for two locations (Mumbai and Delhi)
The different data fields are:

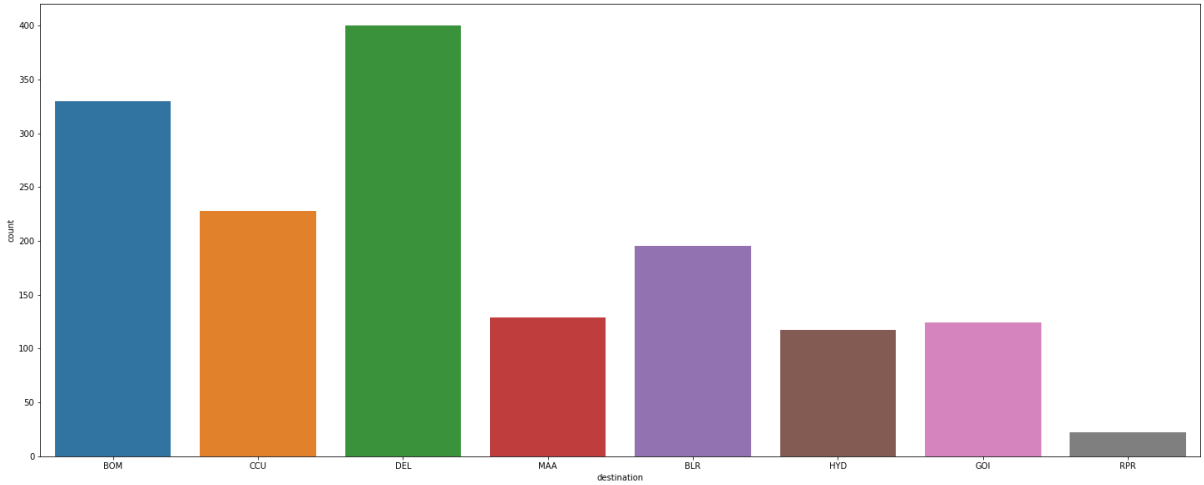
1. Airline name
2. Date of journey
3. source
4. destination
5. route
6. departure time
7. arrival time
8. duration
9. total stops
10. Target variable price

```
Vistara                                366
IndiGo                                 360
Air India                              190
GoAir                                  141
Hahn Air Systems                       135
SpiceJet                               129
AirAsia India                          116
Qatar Airways                           25
Novaturas                               23
Emirates                                18
Singapore Airlines                       9
Vistara + Qatar Airways                  8
Flexflight                               7
Qatar Airways + Vistara                  6
Alliance Air                             2
IndiGo + Vistara                         2
SriLankan Airlines                       2
Qatar Airways + SriLankan Airlines       1
Oman Air                                 1
Qatar Airways                            1
IndiGo                                   1
Air India                                1
AirAsia India                            1
```



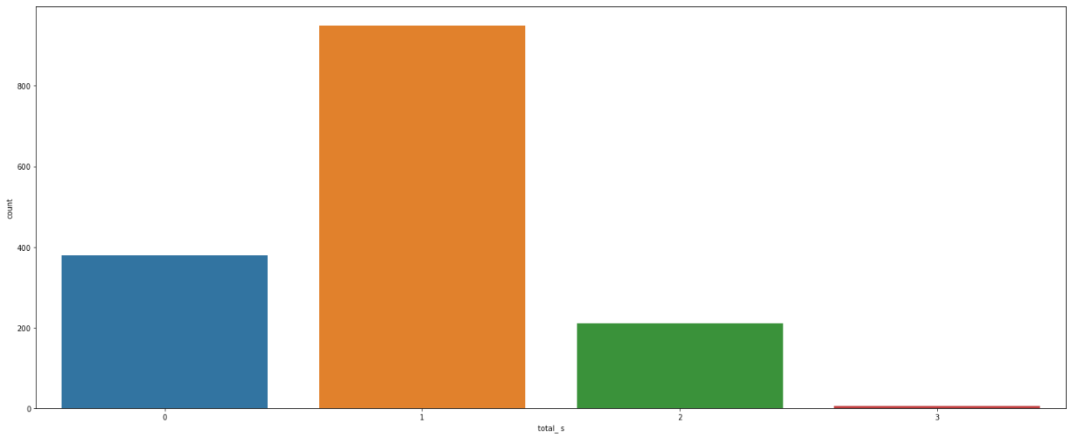
1. Source Type:

```
DEL    914
BLR    292
BOM    279
JAI     41
RPR     19
```



2. Destination

```
DEL    400
BOM    330
CCU    228
BLR    195
MAA    129
GOI    124
HYD    117
RPR     22
```



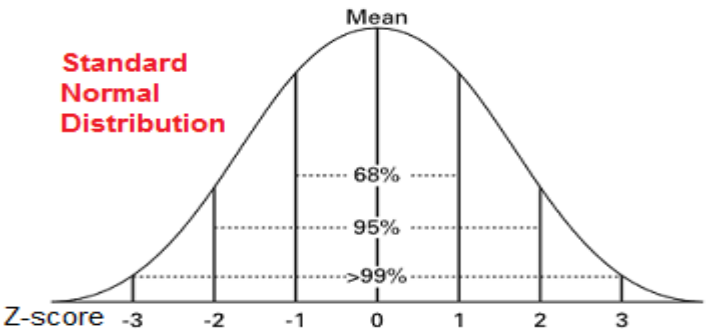
3. Price

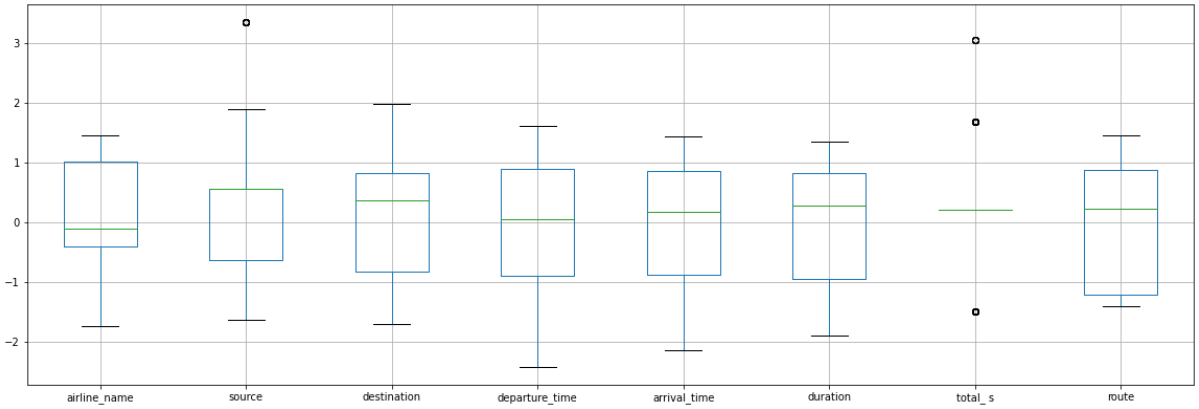
```
1     949
0     379
2     211
3       6
```

- Data Pre-processing Done

```
airline_name                0
source                      0
destination                 0
date_of_journey             0
departure_time              0
arrival_time                0
duration                    0
total_ s                    0
route                     379
target_variable_price       0
dtype: int64
```

- Now as we see there are null values, thus we remove the null values. To remove the null values, for float data types we use mean of remaining data and for object data types we use mode of remaining data.
- Also, the object data types are encoded as then it is easy to operate on the variables.
- Now we need to check whether outliers are present in the data or not. For that we need to check if the z value/z score of all the factors is exceeding the range (-3,3). As we want the data to be normally distributed in the range of (-3,3) as the data be in the 99% domain. Thus, this will be the best data to work with.





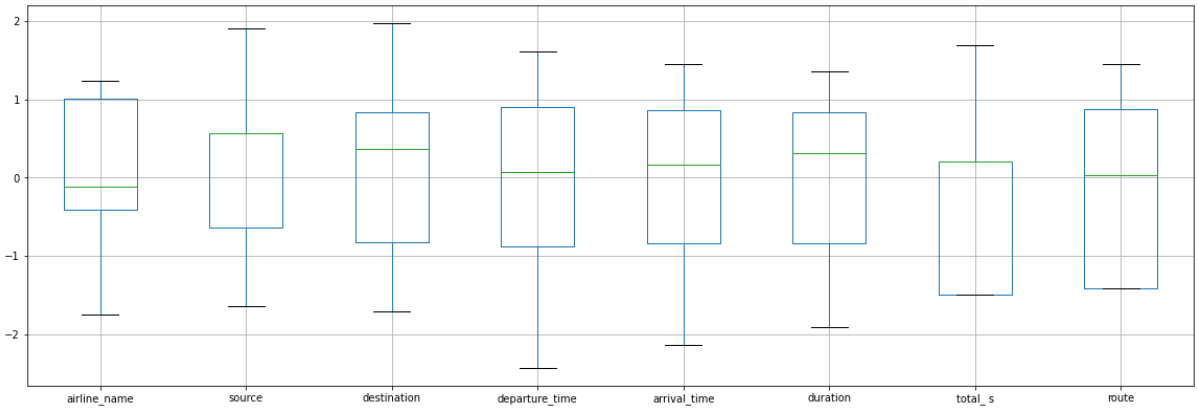
- Here we see there are a few outliers, thus removing them and bringing the data in the range of (-3,3)

  from scipy.stats import zscore

  z=np.abs(zscore(new_df))

  new_df=new_df[(z<3).all(axis=1)]

- Using the above code, we were successfully able to remove the outliers and we get the below range of data



- Now we can proceed with modelling

• Data Inputs- Logic- Output Relationships

Now here we know that there are 10 independent variables which are contributing to predict the dependent variable that is the Flight Price of prediction. Here the relationship is of Linear Regression.

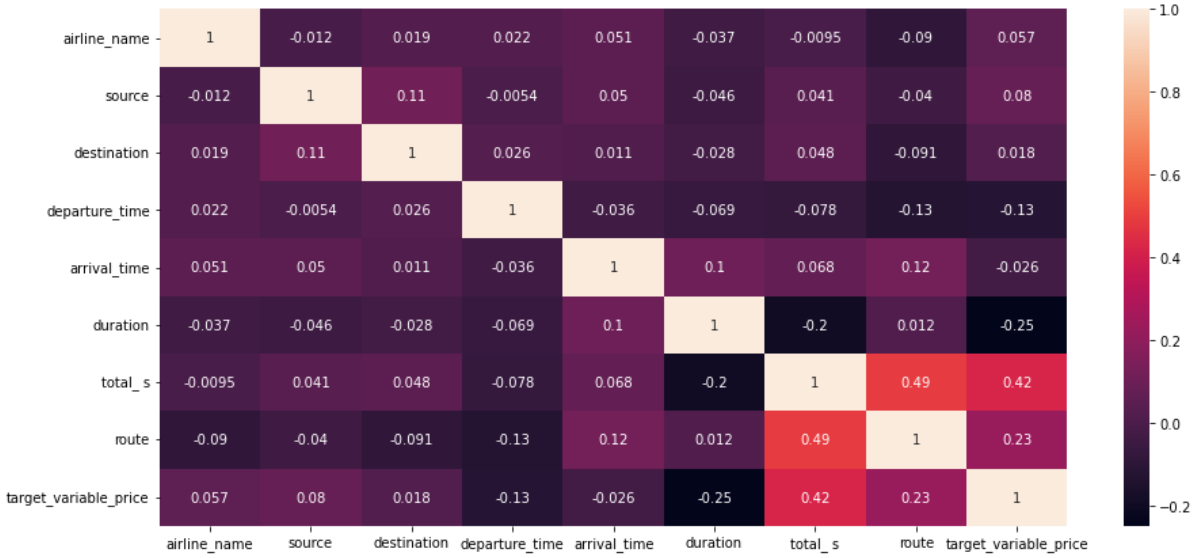
Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.

Thus, assuming they follow a linear regression model, the relationship between the input variables or independent variables and the output variable or dependent variable is:

$$Y_i = f(X_i, \beta) + e_i$$

where Xi is the explanatory/8 independent variables and Yi is the dependent variable/Car Sale Price. f is the function, Beta is the unknown parameters and ei is the error terms.

- Correlation: Correlation is a statistical term describing the degree to which two variables move in coordination with one another. If the two variables move in the same direction, then those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation. Here we see the below relationship.



- Correlation important factors:

```
target_variable_price    1.000000
total_ s                 0.422928
route                    0.225330
source                   0.079861
airline_name             0.057139
destination              0.018346
arrival_time            -0.025582
departure_time          -0.125522
duration                -0.249817
Name: target_variable_price, dtype: float64
```

We see that the factors affecting the Flight Price are Total Stops and Route. Also, the negative factors affecting the Flight Price is the Destination, Arrival Time, Departure Time, Duration Time.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - Cross-validation:

    It is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

  - Hyper Parameter Testing:

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

There are two types of searches used for hyper parameter testing: Research and Randomized Search CV.

With small data sets and lots of resources, Grid Search will produce accurate results. However, with large data sets, the high dimensions will significantly slow down computation time and be very expensive. In this instance, it is advised to use Randomized Search. Thus, here we have used Grid Search CV as the data is too little thus better accuracy.

- Testing of Identified Approaches (Algorithms)

  - First importing all the libraries and models for train test split and linear regression

    from sklearn.model_selection import train_test_split

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()

from sklearn.preprocessing import MinMaxScaler

mms=MinMaxScaler()

from sklearn.metrics import r2_score:
```

- Now testing the random state for maximum accuracy for dividing the data into test and train with the logic of 80:20 which is 80% of data is training data and 20% is testing data

```
for i in range (0,1000) :
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=i)
    lr.fit(x_train,y_train)
    pred_test=lr.predict(x_test)
    pred_train=lr.predict(x_train)
    print(f"at random state {i},the training accuracy is :{r2_score(y_train,pred_train)}")
    print(f"at random state {i},the testing accuracy is :{r2_score(y_test,pred_test)}")
    print("\n")
```

Output:

```
at random state 71, the training accuracy is :0.17473637007993104
at random state 71, the testing accuracy is :0.17747877028269887
```

We chose the random state as 71 as the testing and training accuracy were the closest to 0.1774

- Now diving the training and test data for random state=71

```
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.2,random_state=243)
```

As the total data is of 1467 data points,

Train data=1173

Test data=294

- Now to check which model is fit for the train and test data

```
lr.fit(x_train,y_train)
```

Output:

```
Linear Regression()
```

- Now to calculate the model accuracy

pred_test=lr.predict(x_test)

print(r2_score(y_test,pred_test))

Output:

```
0.17747877028269887
```

Thus, the model accuracy came out to be 17.74%

- Now to test for cross validation or cross fold no

Train_accuracy=r2_score(y_train,pred_train)

Test_accuracy=r2_score(y_test,pred_test)

from sklearn.model_selection import cross_val_score

for j in range (2,10):

cv_score=cross_val_score(lr,x,y,cv=j)

cv_mean=cv_score.mean()

print (f'At cross fold {j} the cv score is {cv_mean} and accuracy score for training is {Train_accuracy}and accuracy score for testing is {Test_accuracy}')

print('\n')

Output:

```
At cross fold 2 the cv score is 0.09598941252090559 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887


At cross fold 3 the cv score is 0.13027677671874952 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887


At cross fold 4 the cv score is 0.12883922496423159 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887


At cross fold 5 the cv score is 0.11730446099862095 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887


At cross fold 6 the cv score is 0.12165424679742785 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887
```

```
At cross fold 7 the cv score is 0.11617600939959472 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887


At cross fold 8 the cv score is -0.009830912486005844 and accuracy scor
e for training is -0.2002703703136668and accuracy score for testing is
0.17747877028269887


At cross fold 9 the cv score is -0.01628703088312886 and accuracy score
for training is -0.2002703703136668and accuracy score for testing is 0.
17747877028269887
```

Here we cross validate that at which cv is the cv score maximum. Thus, we see that at cv=3 the cv score is maximum of 13.02% and the testing accuracy is 20.02% and training accuracy is 17.74% which should ideally be more than the testing accuracy

- Run and evaluate selected models

From the above testing we got the data which needs to be worked upon and that the best fit model is the Linear Regression Model with accuracy of 17.74% and the cv score is 9.69% and the model accuracy improved to 79.07% using the hyper parameter testing

- Lasso Regression

```
from sklearn.model_selection import GridSearchCV

from sklearn.model_selection import cross_val_score

import warnings

warnings.filterwarnings('ignore')

from sklearn.linear_model import Lasso

parameters= {'alpha': [.0001,.001,.01,.1,1,10],'random_state':list(range(0,10))}

ls=Lasso ()

clf=GridSearchCV(ls,parameters)

clf.fit(x_train,y_train)

print(clf.best_params_)
```

Output:

```
{'alpha': 10, 'random_state': 0}
```

ls=Lasso (alpha=10, random_state=0)

ls.fit(x_train,y_train)

ls.score(x_train,y_train)

pred_ls=ls.predict(x_test)

lss=r2_score(y_test,pred_ls)

lss

Output:

```
0.1774786566663632
```

cv_score=cross_val_score(ls,x,y,cv=3)
cv_mean=cv_score.mean()

cv_mean

Output:

```
0.45256293067365183
```

Here we see that the model accuracy is 17.74% and the cross-validation score is 45.25%

- Random Forest Regression

from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor

parameters = {'criterion':['mse','mae'],'max_features':["auto","sqrt","log2"]}
rf=RandomForestRegressor()
clf=GridSearchCV(rf,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)

Output:

```
{'criterion': 'mse', 'max_features': 'auto'}
```

rf=RandomForestRegressor(criterion='mse',max_features=' auto')
rf.fit(x_train,y_train)
rf.score(x_train,y_train)
pred_decision=rf.predict(x_test)

rfs=r2_score(y_test,pred_decision)
*print('R2 score:',rfs*100)*

rfscore=cross_val_score(rf,x,y,cv=2)
rfc=rfscore.mean()
*print('Cross Val Score:',rfc*100)*

Output:

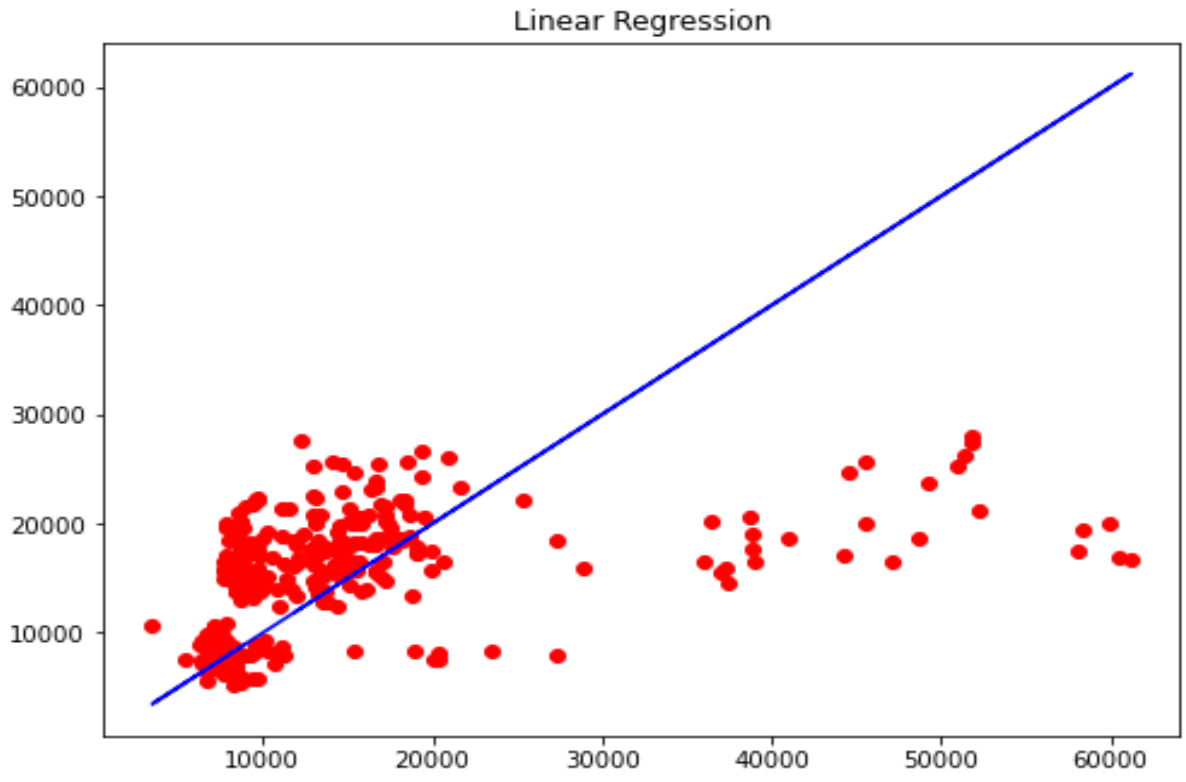
```
R2 score: 79.07587638638779
Cross Val Score: 59.78715333796458
```

- Key Metrics for success in solving problem under consideration

  The key metrics for improving the model accuracy are:
  Cross Validation
  Hyper Parameter testing

- Visualizations

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_test,color='r')
plt.plot(y_test,y_test,color='b')
plt.title("Linear Regression")
plt.show()
```



- Interpretation of the Results

  Here we see that our data points are not in line with the blue line which is the ideal line or slope of the linear regression equation. Thus, this shows that the model is not perfectly fitted, we need to work on the model as the data we have taken is biased on locations. Here the model accuracy is almost the same to 80% but the cross-validation score is increased to 60% which is not so close to the model accuracy which means that the model is overfitted.

CONCLUSION

- Key Findings and Conclusions of the Study

  We did Exploratory information Investigation on the highlights of this dataset and saw how each include is distributed.

  We dissected each variable to check in the event that information is cleaned and ordinarily distributed. We cleaned
  the information and evacuated NA values. We tried to find the correlation and based on the outcomes, we accepted whether or not there's a relation between the Price of the flight and the other factors, we see that the positive factors affecting the Flight Price are Total Stops and Route, But the most important positively affecting factor is Total Stops as it with increase in the stops, Flight price is increasing. Also, the negative factors affecting the Flight price are Duration, Departure time, arrival time. We used the linear regression model to predict the Flight price .

  We got a model with an 80% accuracy which is good. Just a disadvantage that the data available to train the model is biased as the data is scraped from one particular website- skyscanner.com. Data quality can be improved. The more accurate the data the better the model will be trained.

  This model can be used people travelling to understand how exactly the prices vary with the variables. They can accordingly plan there journey with the received data . Further, the model will be a good

way for the management to understand the pricing dynamics of a new market.

- Learning Outcomes of the Study in respect of Data Science

- Able to demonstrate proficiency with statistical analysis of data.
- Ability to build and assess data-based models.
- Data management.
- Able to do basic data cleaning, and can transform variables to facilitate analysis.
- How variables are connected and changing the independent variables how the dependent variable changes.
- To perform hyper parameter testing and cross validation.
- Interpretation of data and results
- How to reach the outputs