FLIP ROBO

FLIGHT PRICE PREDICTION PROJECT

Submitted by:
Tarkeshwer Pandey

ACKNOWLEDGEMENT

It is a great pleasure to express my gratitude to Flip Robo, for giving me the opportunity to work on an interesting project, which helped me in improving my knowledge, coding skills and my analyzation skills.

Flip Robo also gave me opportunity to build PowerPoint Presentation and Project Report, which will help me to share steps taken while building the entire model. It has helped me in deciding about the future prospects of various Data Science fields. Now, I will explain the understanding of the project through this report.

Introduction of Project: -

The tourism industry is changing fast and this is attracting a lot more travelers each year. The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Nowadays, flight prices are quite unpredictable. The ticket prices change frequently.

Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible. Using technology, it is actually possible to reduce the uncertainty of flight prices. So here we will be predicting the flight prices using efficient machine learning techniques.

Airline: The Name of flight.

Travel Date: The date when the journey starts from the source.

From: From Which destination to fly. To: The destination where to arrive

Dep_Time: - Time when the flight takes off.

Arrival_Time: - Time when the flight arrives at thedestination.

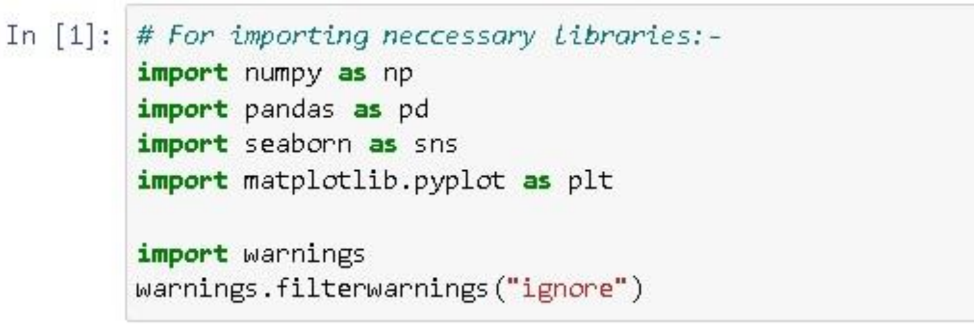
Stops: - Number of layovers in between reaching destination.Price: The price of the ticket.

MOTIVATION FOR THE PROBLEM UNDERTAKEN: -

For Modelling this dataset, Flight Price Prediction with all given available independent variables. This model will then be used for management of how the customer will be able to spend money on high priced tickets based on the independent variables. With the help of this prediction model, it will be decided accordingly and manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be prediction based insights to the management to understand whether the customer will pay the suitable price as compared to high priced flight Fares.

Importing Libraries: -

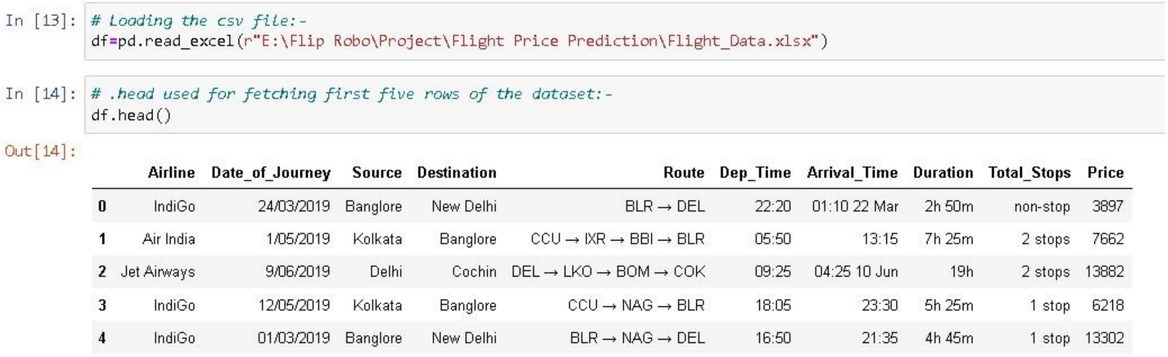
Here, we are importing all the libraries which are required for EDA, visualization, prediction and finding all matrics. The reason of doing this is that it become easier to use all the import statement at one go and we do not require to import the statement again at each point.



```
In [1]: # For importing neccessary libraries:-
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt

        import warnings
        warnings.filterwarnings("ignore")
```

Data Sources and their Formats: -

Now I am going to upload or read the files/datasets using pandas. For this I have used read_csv method:-



```
In [13]: # Loading the csv file:-
         df=pd.read_excel(r"E:\Flip Robo\Project\Flight Price Prediction\Flight_Data.xlsx")

In [14]: # .head used for fetching first five rows of the dataset:-
         df.head()

Out[14]:
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	13302

Some EDA steps:-

1. For checking the rows and columns present in the dataset

   Command Used:- **data.shape**

2. For Checking the null values in the dataset:-

   Command used:- **data.isnull().sum()**

3. For checking the available columns in the dataset:

   Command used:- **data.columns**

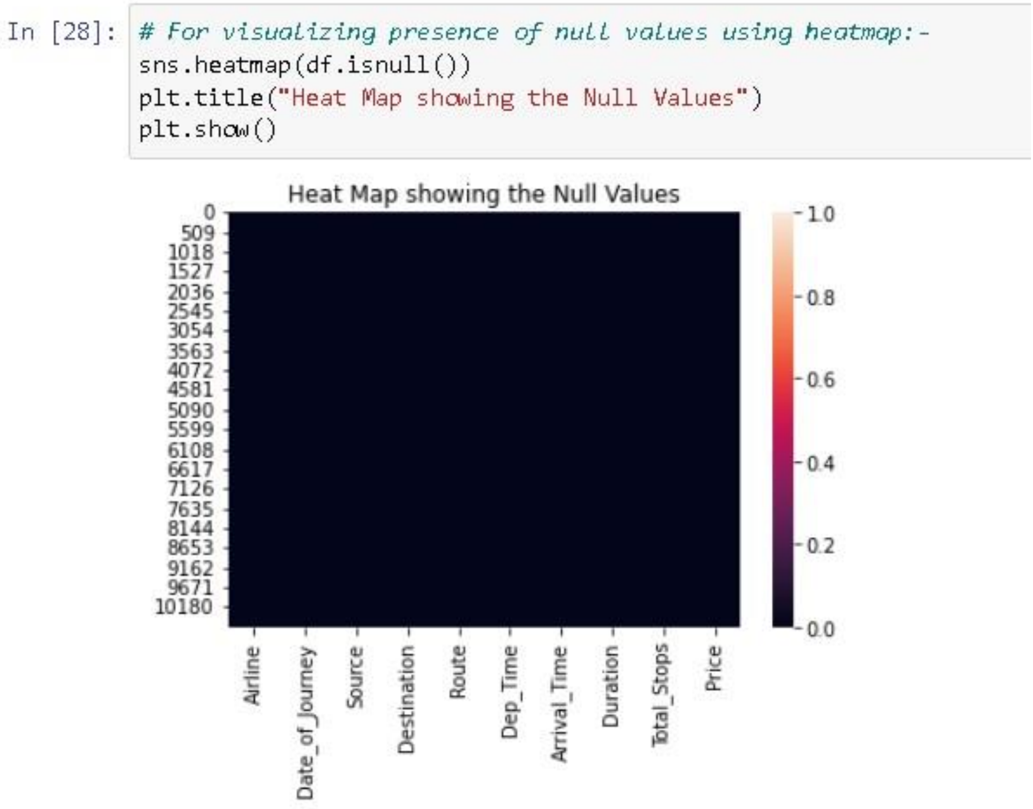
4. FOR CHECKING THE DATATYPE OF EACH FEATURES:-

   Command Used:- data.dtypes()

5. FOR OBSERVING THE INFORMATION ABOUT DATASET:-

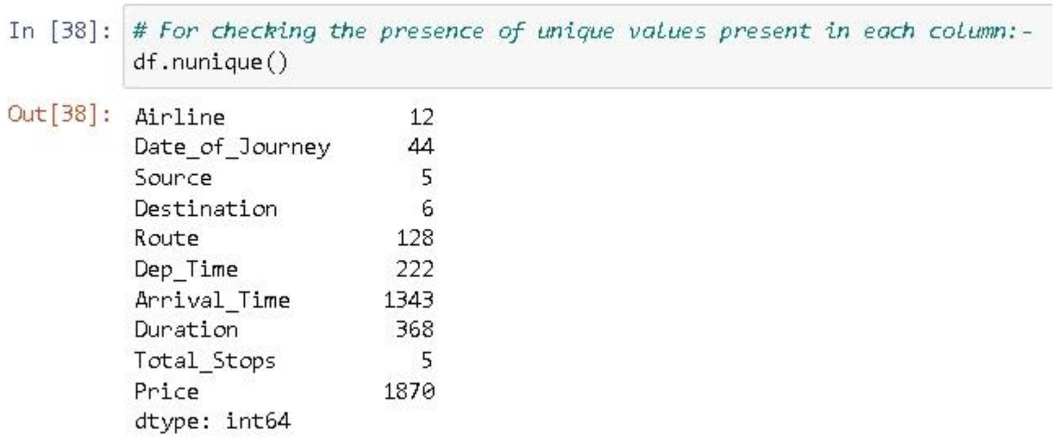
   Command Used :- data.info()

DATA VISUALISATIONS: -



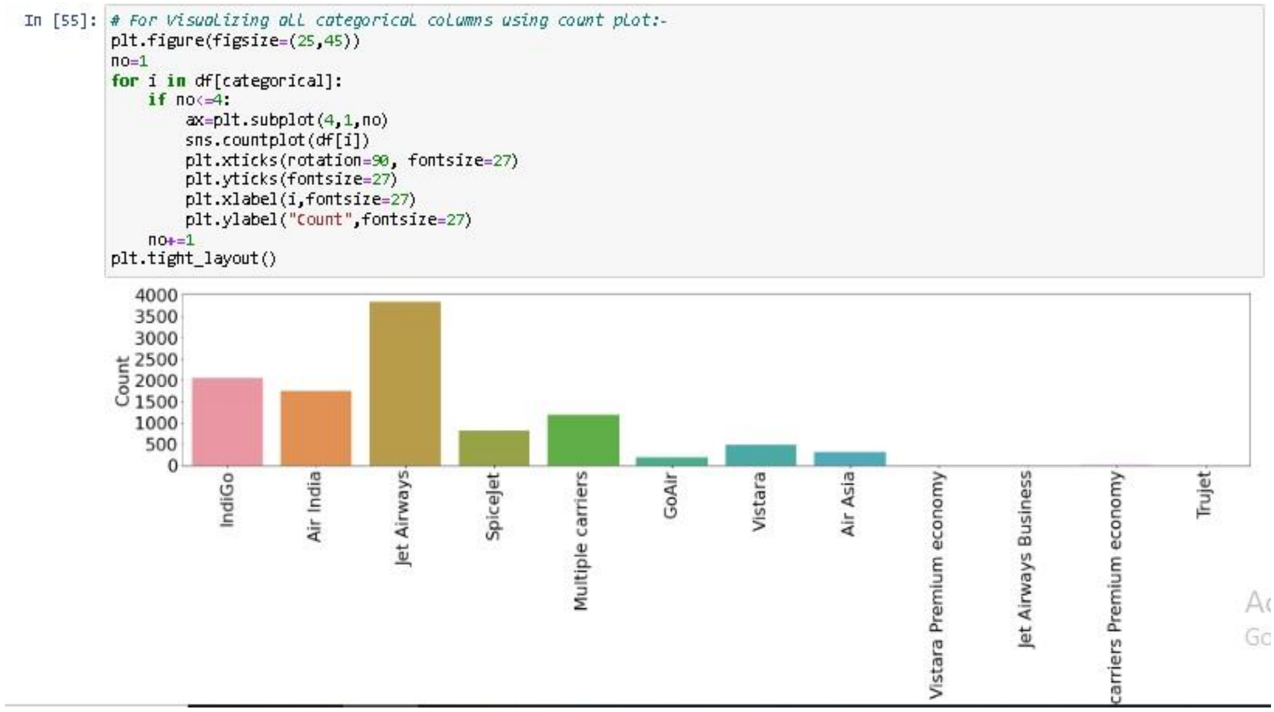
```
In [28]: # For visualizing presence of null values using heatmap:-
         sns.heatmap(df.isnull())
         plt.title("Heat Map showing the Null Values")
         plt.show()
```

For Unique Values: -



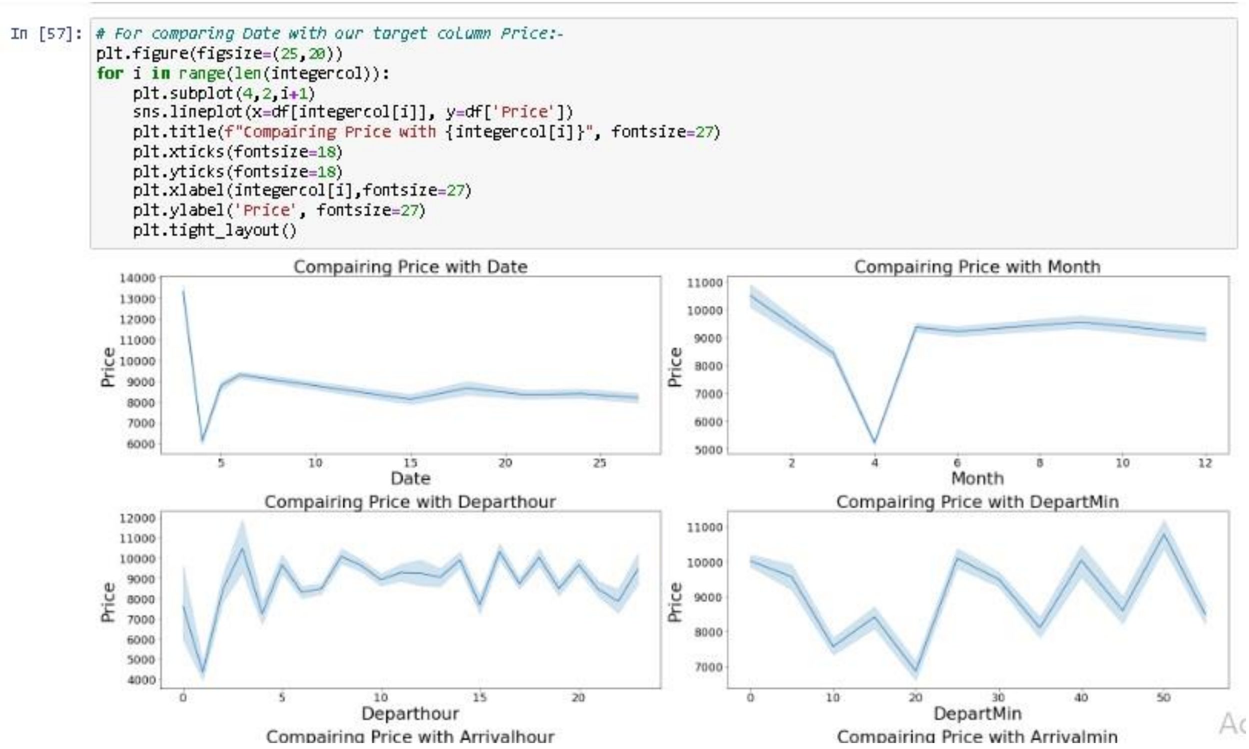
```
In [38]: # For checking the presence of unique values present in each column:-
         df.nunique()

Out[38]: Airline              12
         Date_of_Journey      44
         Source                5
         Destination           6
         Route               128
         Dep_Time            222
         Arrival_Time       1343
         Duration            368
         Total_Stops           5
         Price              1870
         dtype: int64
```

For Visualizing Categorical columns: -



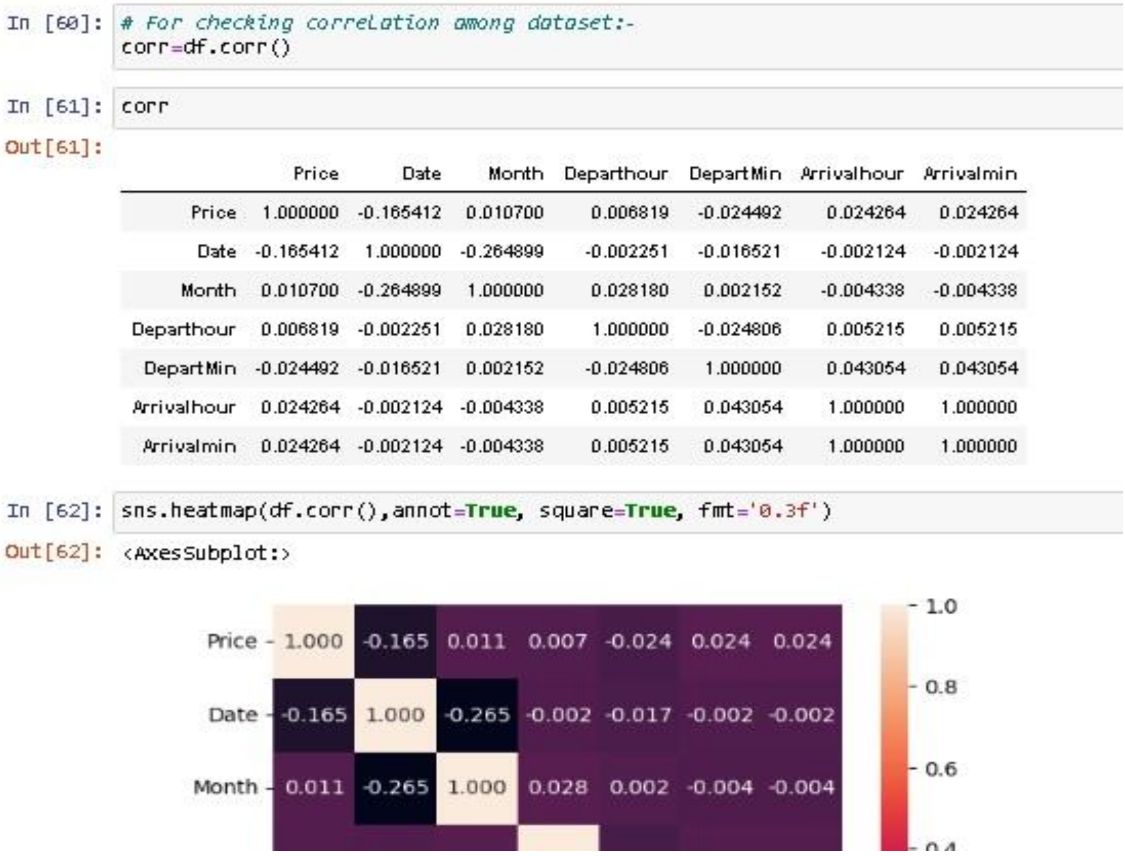
```
In [55]: # For visualizing all categorical columns using count plot:-
         plt.figure(figsize=(25,45))
         no=1
         for i in df[categorical]:
             if no<=4:
                 ax=plt.subplot(4,1,no)
                 sns.countplot(df[i])
                 plt.xticks(rotation=90, fontsize=27)
                 plt.yticks(fontsize=27)
                 plt.xlabel(i,fontsize=27)
                 plt.ylabel("Count",fontsize=27)
             no+=1
         plt.tight_layout()
```

For Comparing Date with Target column: -



```
In [57]: # For comparing Date with our target column Price:-
         plt.figure(figsize=(25,20))
         for i in range(len(integercol)):
             plt.subplot(4,2,i+1)
             sns.lineplot(x=df[integercol[i]], y=df['Price'])
             plt.title(f"Compairing Price with {integercol[i]}", fontsize=27)
             plt.xticks(fontsize=18)
             plt.yticks(fontsize=18)
             plt.xlabel(integercol[i],fontsize=27)
             plt.ylabel('Price', fontsize=27)
             plt.tight_layout()
```

Correlation: -



In [60]:
```
# For checking correlation among dataset:-
corr=df.corr()
```

In [61]:
```
corr
```

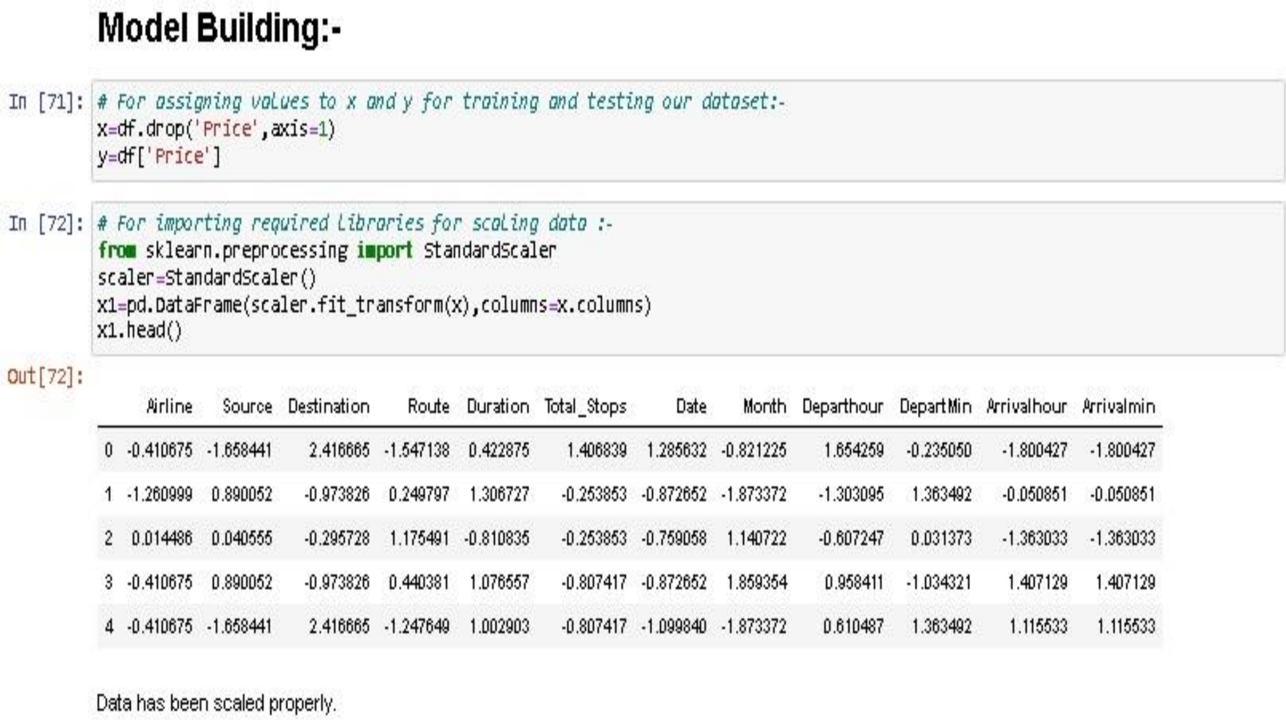
Out[61]:

	Price	Date	Month	Departhour	DepartMin	Arrivalhour	Arrivalmin
Price	1.000000	-0.165412	0.010700	0.006819	-0.024492	0.024264	0.024264
Date	-0.165412	1.000000	-0.264899	-0.002251	-0.016521	-0.002124	-0.002124
Month	0.010700	-0.264899	1.000000	0.028180	0.002152	-0.004338	-0.004338
Departhour	0.006819	-0.002251	0.028180	1.000000	-0.024806	0.005215	0.005215
DepartMin	-0.024492	-0.016521	0.002152	-0.024806	1.000000	0.043054	0.043054
Arrivalhour	0.024264	-0.002124	-0.004338	0.005215	0.043054	1.000000	1.000000
Arrivalmin	0.024264	-0.002124	-0.004338	0.005215	0.043054	1.000000	1.000000

In [62]:
```
sns.heatmap(df.corr(),annot=True, square=True, fmt='0.3f')
```

Out[62]: `<AxesSubplot:>`

Model Building Phase: -



Model Building:-

In [71]:
```
# For assigning values to x and y for training and testing our dataset:-
x=df.drop('Price',axis=1)
y=df['Price']
```

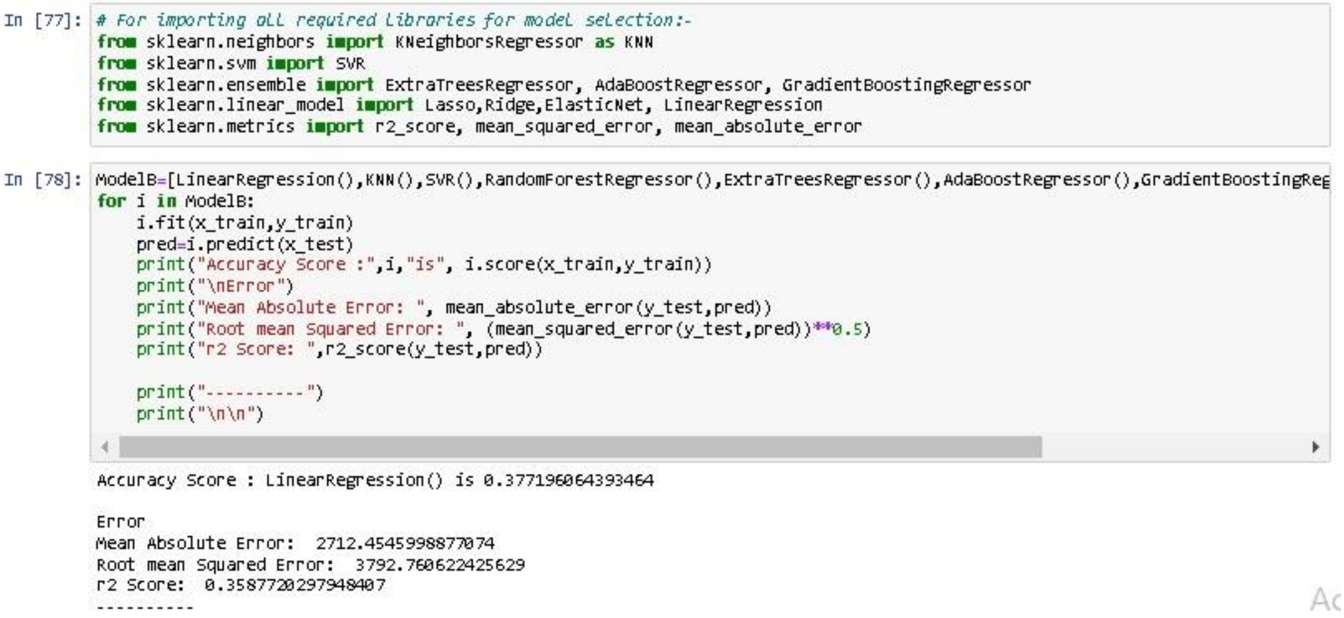
In [72]:
```
# For importing required libraries for scaling data :-
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x1=pd.DataFrame(scaler.fit_transform(x),columns=x.columns)
x1.head()
```

Out[72]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Date	Month	Departhour	DepartMin	Arrivalhour	Arrivalmin
0	-0.410675	-1.658441	2.416665	-1.547138	0.422875	1.406839	1.285632	-0.821225	1.654259	-0.235050	-1.800427	-1.800427
1	-1.260999	0.890052	-0.973826	0.249797	1.306727	-0.253853	-0.872652	-1.873372	-1.303095	1.363492	-0.050851	-0.050851
2	0.014486	0.040555	-0.295728	1.175491	-0.810835	-0.253853	-0.759058	1.140722	-0.607247	0.031373	-1.363033	-1.363033
3	-0.410675	0.890052	-0.973826	0.440381	1.076557	-0.807417	-0.872652	1.859354	0.958411	-1.034321	1.407129	1.407129
4	-0.410675	-1.658441	2.416665	-1.247649	1.002903	-0.807417	-1.099840	-1.873372	0.610487	1.363492	1.115533	1.115533

Data has been scaled properly.

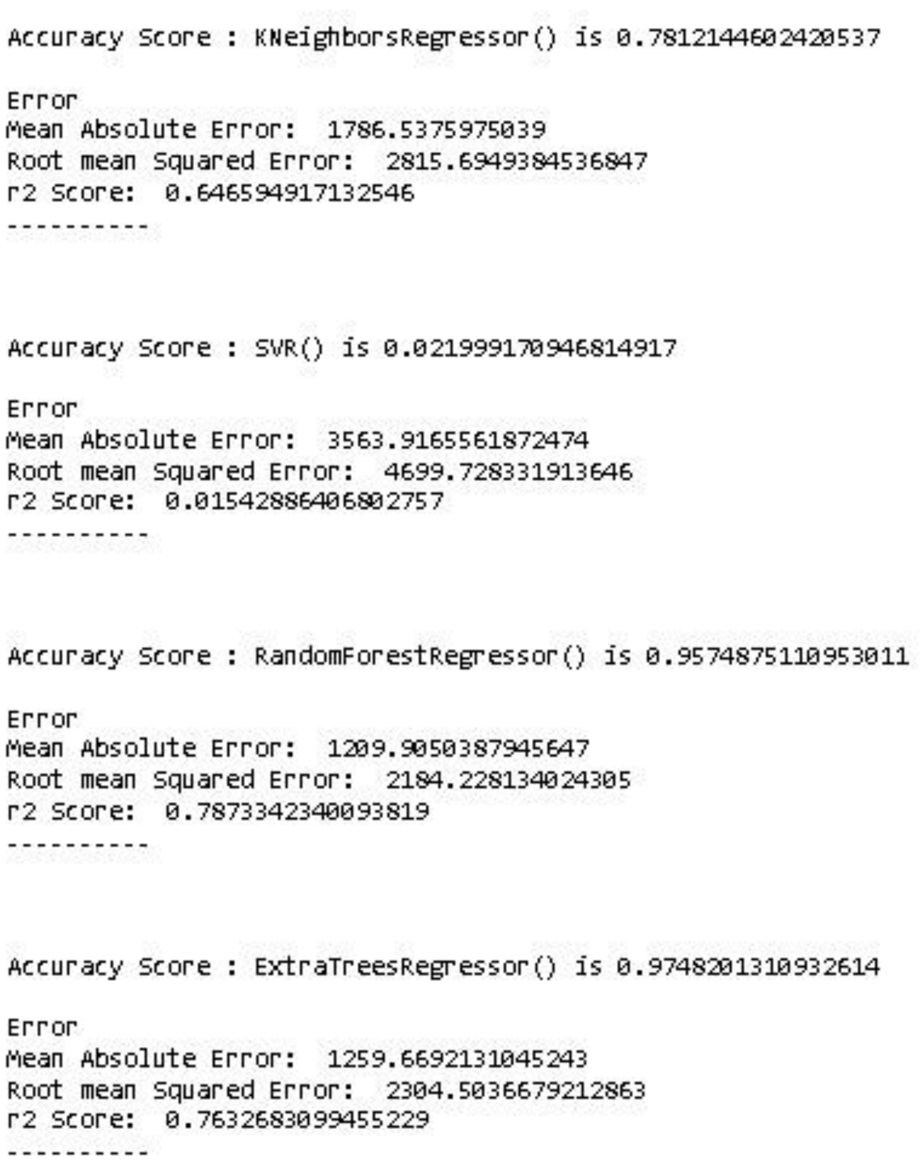
Different Model Scores: -



```
In [77]: # For importing all required libraries for model selection:-
         from sklearn.neighbors import KNeighborsRegressor as KNN
         from sklearn.svm import SVR
         from sklearn.ensemble import ExtraTreesRegressor, AdaBoostRegressor, GradientBoostingRegressor
         from sklearn.linear_model import Lasso,Ridge,ElasticNet, LinearRegression
         from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

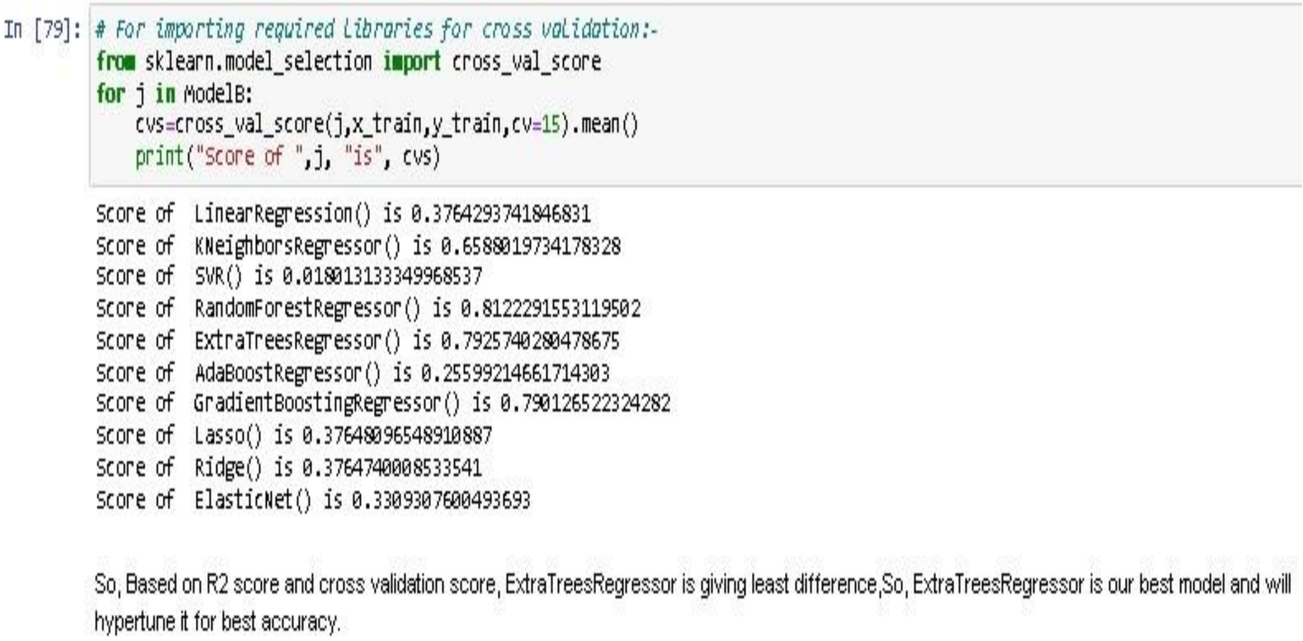
```
In [78]: ModelB=[LinearRegression(),KNN(),SVR(),RandomForestRegressor(),ExtraTreesRegressor(),AdaBoostRegressor(),GradientBoostingReg
         for i in ModelB:
             i.fit(x_train,y_train)
             pred=i.predict(x_test)
             print("Accuracy Score :",i,"is", i.score(x_train,y_train))
             print("\nError")
             print("Mean Absolute Error: ", mean_absolute_error(y_test,pred))
             print("Root mean Squared Error: ", (mean_squared_error(y_test,pred))**0.5)
             print("r2 Score: ",r2_score(y_test,pred))

             print("----------")
             print("\n\n")
```

```
Accuracy Score : LinearRegression() is 0.377196064393464

Error
Mean Absolute Error:  2712.4545998877074
Root mean Squared Error:  3792.760622425629
r2 Score:  0.3587720297948407
----------
```



```
Accuracy Score : KNeighborsRegressor() is 0.7812144602420537

Error
Mean Absolute Error:  1786.5375975039
Root mean Squared Error:  2815.6949384536847
r2 Score:  0.646594917132546
----------



Accuracy Score : SVR() is 0.021999170946814917

Error
Mean Absolute Error:  3563.9165561872474
Root mean Squared Error:  4699.728331913646
r2 Score:  0.015428864068802757
----------



Accuracy Score : RandomForestRegressor() is 0.9574875110953011

Error
Mean Absolute Error:  1209.9050387945647
Root mean Squared Error:  2184.228134024305
r2 Score:  0.7873342340093819
----------



Accuracy Score : ExtraTreesRegressor() is 0.9748201310932614

Error
Mean Absolute Error:  1259.6692131045243
Root mean Squared Error:  2304.5036679212863
r2 Score:  0.7632683099455229
----------
```

Cross Validation Phase:-



```
In [79]: # For importing required libraries for cross validation:-
         from sklearn.model_selection import cross_val_score
         for j in ModelB:
             cvs=cross_val_score(j,x_train,y_train,cv=15).mean()
             print("Score of ",j, "is", cvs)
```

```
Score of  LinearRegression() is 0.3764293741846831
Score of  KNeighborsRegressor() is 0.6588019734178328
Score of  SVR() is 0.018013133349968537
Score of  RandomForestRegressor() is 0.8122291553119502
Score of  ExtraTreesRegressor() is 0.7925740280478675
Score of  AdaBoostRegressor() is 0.25599214661714303
Score of  GradientBoostingRegressor() is 0.790126522324282
Score of  Lasso() is 0.37648096548910887
Score of  Ridge() is 0.3764740008533541
Score of  ElasticNet() is 0.3309307600493693
```

So, Based on R2 score and cross validation score, ExtraTreesRegressor is giving least difference,So, ExtraTreesRegressor is our best model and will hypertune it for best accuracy.

Interpretation of the Results: -

1. I have used visualization tools such as dist Plot, Count Plot for categorical data and line plot to understand thedata in a better way.
2. I have done the model building process with several algorithms and the best model is Extra Trees Regressorwith an accuracy score of 71% after Hyper Parameter Tuning.

CONCLUSION: -

The overall survey for the dynamic price changes in the flight tickets is presented. This gives the information about the ups and downs in the airfares according to the days, weekend and time of the day that is morning, evening and night. Also, the machine learning models in the computational intelligence field that are evaluated before on different datasets are studied. Their accuracy and performances are evaluated and compared in order to get better result. For the prediction of the ticket prices perfectly different prediction models are tested for the better prediction accuracy. As the pricing models of the company are developed in order to maximize the revenue management, so to get the result with maximum accuracy regression analysis is used. From the studies, the feature that influences the prices of the ticket are to be considered. In future, the details about number of available seats can improve the performance of the model

***

Thank you