FLIGHT PRICE PREDICTION PROJECT

Name of Data Scientist: Vaibhav Tayade

Contact details: vaibhav_t29@rediffmail.com

Project submitted as a part of internship projects

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, we 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.

Phase 1: In this phase we'll work on collecting flight journey datasets from website like easemytrip.com

We will here use technique of web scraping using various libraries like beautiful soup and selenium library in order to collect dataset.

Phase 2: In this phase we'll work on 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.

INTRODUCTION:

Booking the flight ticket is the big task for the frequent goers, because of the dynamic nature of the flight booking websites, it’s become very much tough for the passenger to book their ticket at affordable price. There are various website available in the market like yatra.com, makemytrip.com. skyscanner.com, easemytrip.com etc. All website are dynamic in nature. In this project we’ll try to work on flight price prediction machine learning model so we’ll be able to predict the price of the flight using various other information.

Phase 1: Web Scraping

In this phase we have used **“http://www.easemytrip.com”** website to collect all flight datasets.

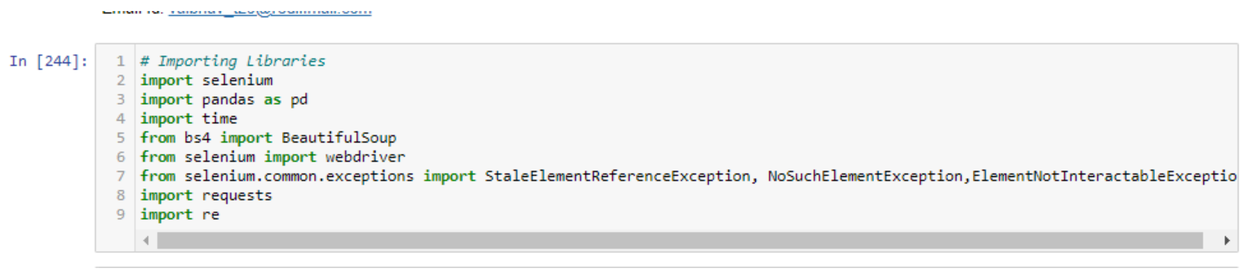
We will try to collect data like –

1) Date of the journey
2) Month of the journey
3) Airway carrier
4) Flight departure timings (in hours and minutes)
5) Flight destination arrival timings (in hours and minutes)
6) From City
7) To City
8) Flight Price

We’ll use beautifulsoup and selenium both the web scraping libraries for collecting the dataset.

Web scrapping code is given below:

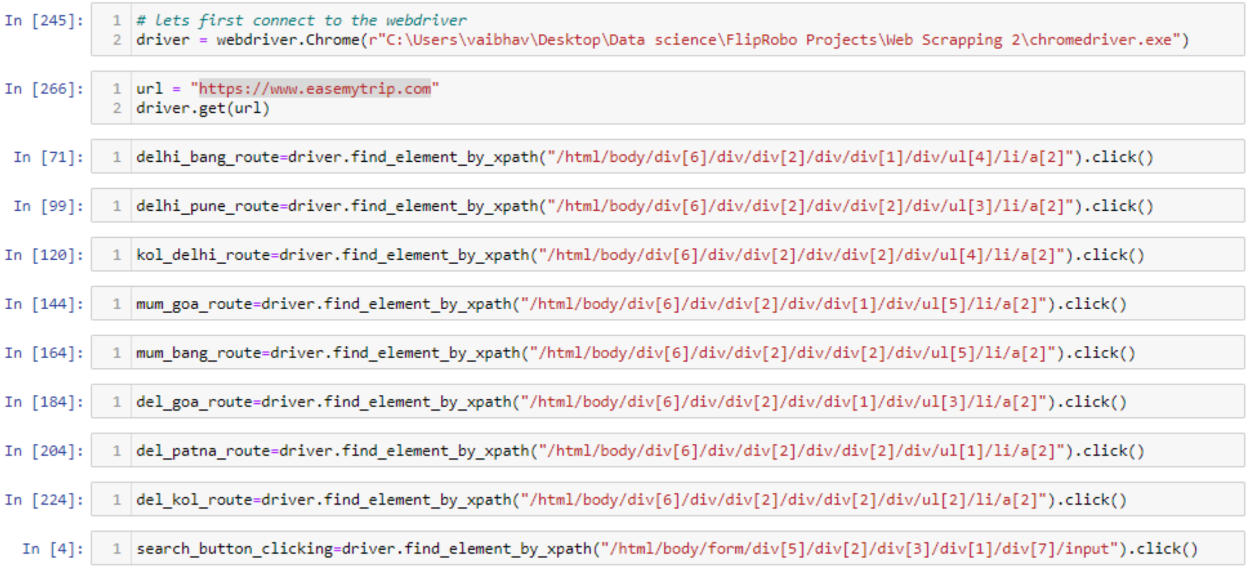
Importing various web scrapping libraries.



In [244]:

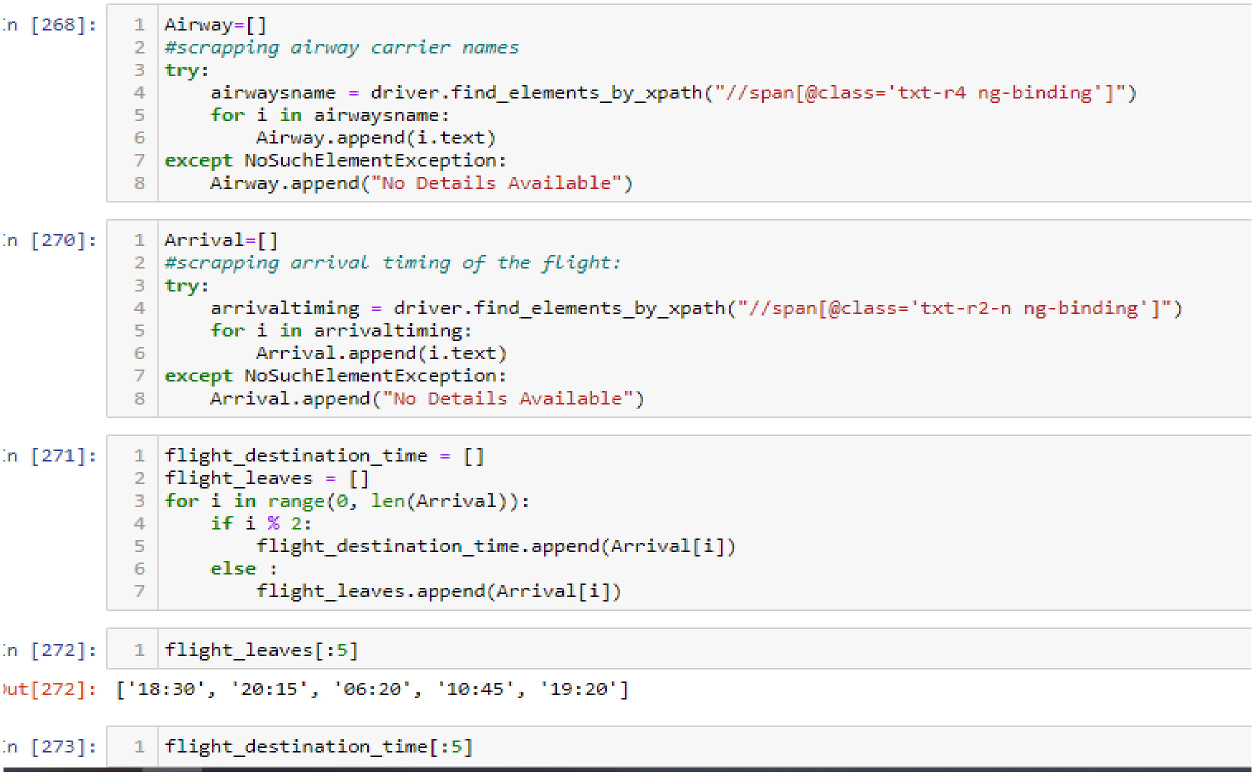
```
# Importing Libraries
import selenium
import pandas as pd
import time
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException,ElementNotInteractableExceptio
import requests
import re
```

Flight routes:



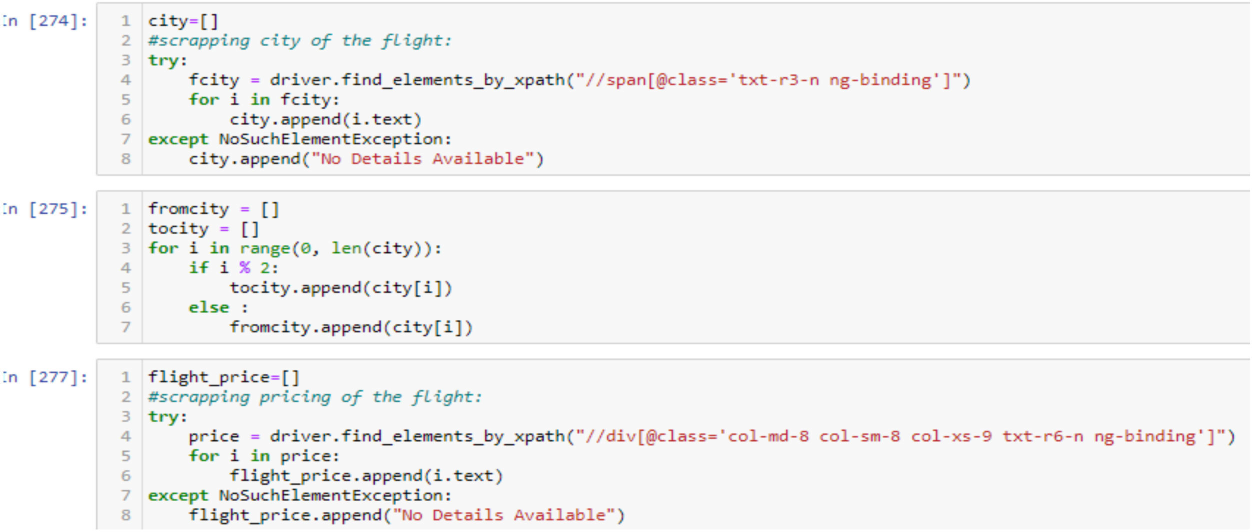
```
In [245]:  # lets first connect to the webdriver
           driver = webdriver.Chrome(r"C:\Users\vaibhav\Desktop\Data science\FlipRobo Projects\Web Scrapping 2\chromedriver.exe")

In [266]:  url = "https://www.easemytrip.com"
           driver.get(url)

In [71]:   delhi_bang_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[1]/div/ul[4]/li/a[2]").click()

In [99]:   delhi_pune_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[2]/div/ul[3]/li/a[2]").click()

In [120]:  kol_delhi_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[2]/div/ul[4]/li/a[2]").click()

In [144]:  mum_goa_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[1]/div/ul[5]/li/a[2]").click()

In [164]:  mum_bang_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[2]/div/ul[5]/li/a[2]").click()

In [184]:  del_goa_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[1]/div/ul[3]/li/a[2]").click()

In [204]:  del_patna_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[2]/div/ul[1]/li/a[2]").click()

In [224]:  del_kol_route=driver.find_element_by_xpath("/html/body/div[6]/div/div[2]/div/div[2]/div/ul[2]/li/a[2]").click()

In [4]:    search_button_clicking=driver.find_element_by_xpath("/html/body/form/div[5]/div[2]/div[3]/div[1]/div[7]/input").click()
```

Collecting dataset:



```
In [268]:  Airway=[]
           #scrapping airway carrier names
           try:
               airwaysname = driver.find_elements_by_xpath("//span[@class='txt-r4 ng-binding']")
               for i in airwaysname:
                   Airway.append(i.text)
           except NoSuchElementException:
               Airway.append("No Details Available")

In [270]:  Arrival=[]
           #scrapping arrival timing of the flight:
           try:
               arrivaltiming = driver.find_elements_by_xpath("//span[@class='txt-r2-n ng-binding']")
               for i in arrivaltiming:
                   Arrival.append(i.text)
           except NoSuchElementException:
               Arrival.append("No Details Available")

In [271]:  flight_destination_time = []
           flight_leaves = []
           for i in range(0, len(Arrival)):
               if i % 2:
                   flight_destination_time.append(Arrival[i])
               else :
                   flight_leaves.append(Arrival[i])

In [272]:  flight_leaves[:5]

Out[272]:  ['18:30', '20:15', '06:20', '10:45', '19:20']

In [273]:  flight_destination_time[:5]
```

Collecting dataset.

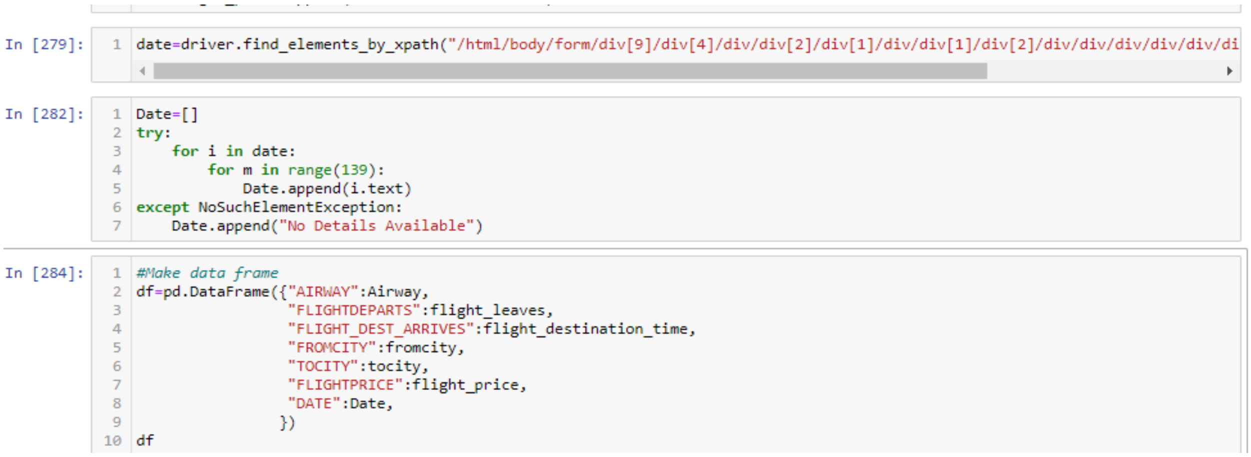


In [274]:
```
city=[]
#scrapping city of the flight:
try:
    fcity = driver.find_elements_by_xpath("//span[@class='txt-r3-n ng-binding']")
    for i in fcity:
        city.append(i.text)
except NoSuchElementException:
    city.append("No Details Available")
```

In [275]:
```
fromcity = []
tocity = []
for i in range(0, len(city)):
    if i % 2:
        tocity.append(city[i])
    else :
        fromcity.append(city[i])
```

In [277]:
```
flight_price=[]
#scrapping pricing of the flight:
try:
    price = driver.find_elements_by_xpath("//div[@class='col-md-8 col-sm-8 col-xs-9 txt-r6-n ng-binding']")
    for i in price:
        flight_price.append(i.text)
except NoSuchElementException:
    flight_price.append("No Details Available")
```

Collecting datasets.

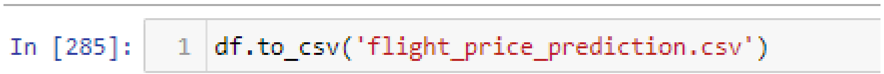


In [279]:
```
date=driver.find_elements_by_xpath("/html/body/form/div[9]/div[4]/div/div[2]/div[1]/div/div[1]/div[2]/div/div/div/div/div/di
```

In [282]:
```
Date=[]
try:
    for i in date:
        for m in range(139):
            Date.append(i.text)
except NoSuchElementException:
    Date.append("No Details Available")
```

In [284]:
```
#Make data frame
df=pd.DataFrame({"AIRWAY":Airway,
                 "FLIGHTDEPARTS":flight_leaves,
                 "FLIGHT_DEST_ARRIVES":flight_destination_time,
                 "FROMCITY":fromcity,
                 "TOCITY":tocity,
                 "FLIGHTPRICE":flight_price,
                 "DATE":Date,
                })
df
```

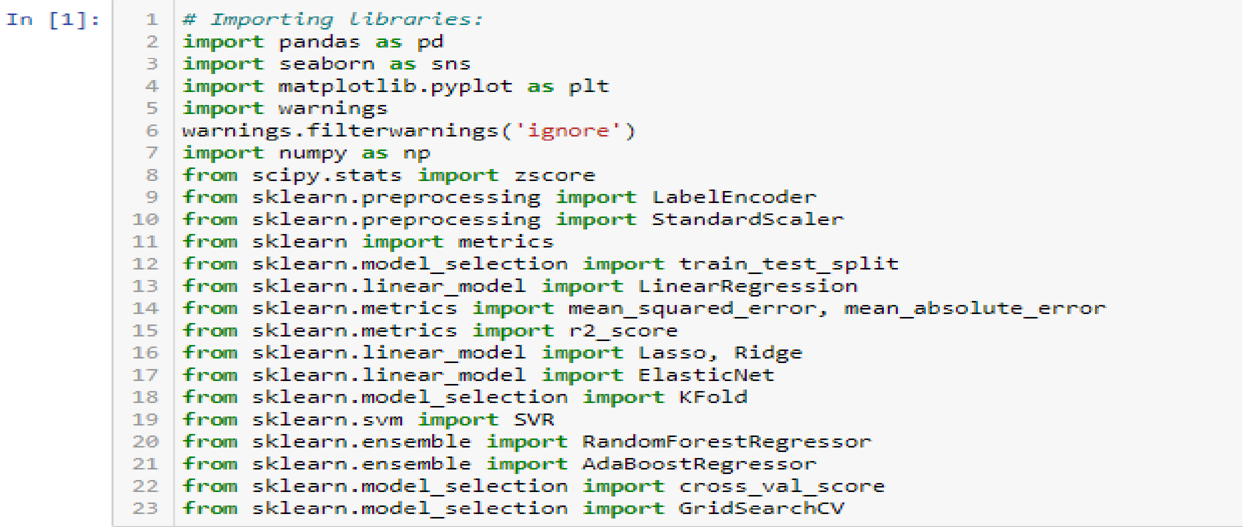
Creating dataframe and saving to csv file.



In [285]:
```
df.to_csv('flight_price_prediction.csv')
```

Phase 2: Machine learning model making:

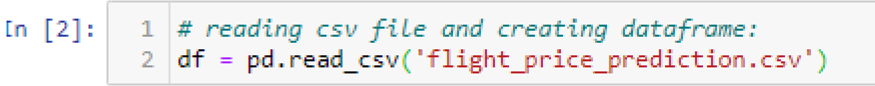
The very first step which we do in making machine learning model is importing various libraries.



In [1]:

```
# Importing libraries:
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import numpy as np
from scipy.stats import zscore
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.linear_model import Lasso, Ridge
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import KFold
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
```

Reading csv datafile:



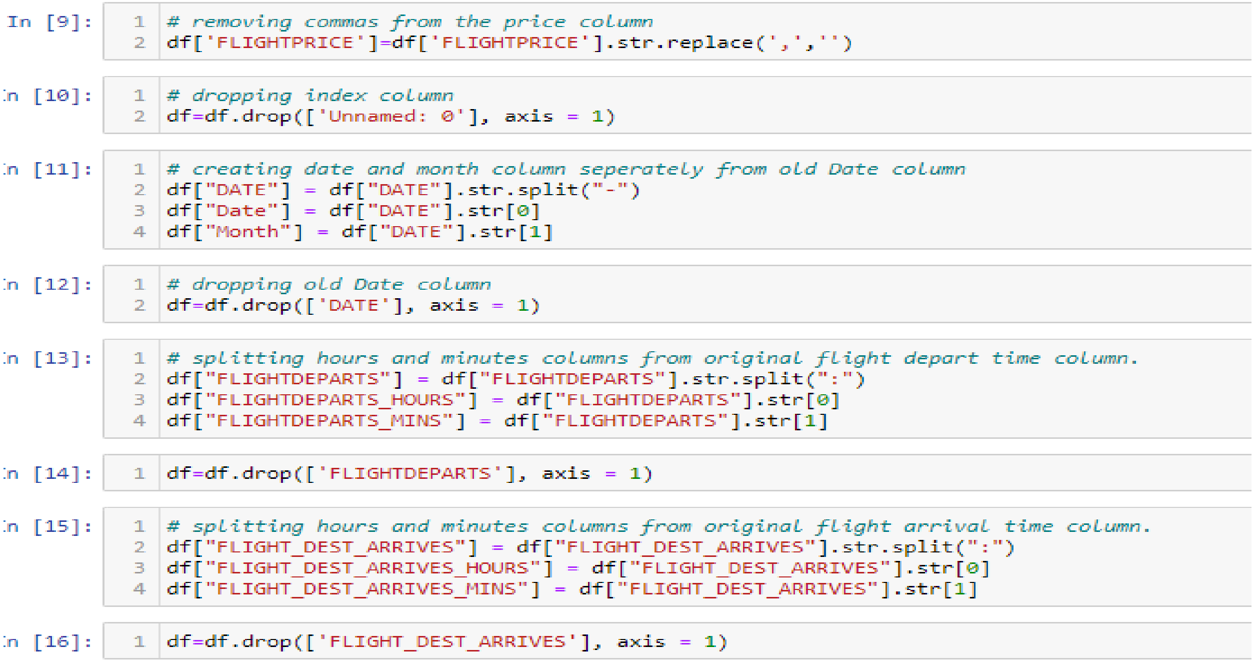
In [2]:

```
# reading csv file and creating dataframe:
df = pd.read_csv('flight_price_prediction.csv')
```

DATA CLEANING PROCESS:

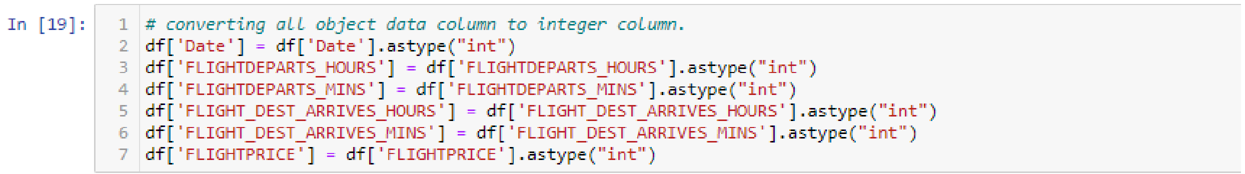
Data Cleaning process is one of the most important process in machine learning model building projects, we need to remove null values, repeated values, deal with string type and numerical data types with respect to dataset.

Here in Flight price column, we have data values in strings data types, which is having ‘commas’ and, we need to deal with it, we need to convert it to pure numerical data type. We have done that with the following code.



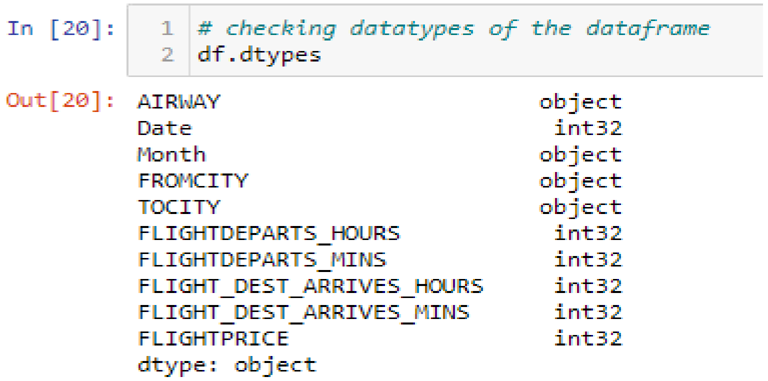
```
In [9]:   # removing commas from the price column
          df['FLIGHTPRICE']=df['FLIGHTPRICE'].str.replace(',','')

In [10]:  # dropping index column
          df=df.drop(['Unnamed: 0'], axis = 1)

In [11]:  # creating date and month column seperately from old Date column
          df["DATE"] = df["DATE"].str.split("-")
          df["Date"] = df["DATE"].str[0]
          df["Month"] = df["DATE"].str[1]

In [12]:  # dropping old Date column
          df=df.drop(['DATE'], axis = 1)

In [13]:  # splitting hours and minutes columns from original flight depart time column.
          df["FLIGHTDEPARTS"] = df["FLIGHTDEPARTS"].str.split(":")
          df["FLIGHTDEPARTS_HOURS"] = df["FLIGHTDEPARTS"].str[0]
          df["FLIGHTDEPARTS_MINS"] = df["FLIGHTDEPARTS"].str[1]

In [14]:  df=df.drop(['FLIGHTDEPARTS'], axis = 1)

In [15]:  # splitting hours and minutes columns from original flight arrival time column.
          df["FLIGHT_DEST_ARRIVES"] = df["FLIGHT_DEST_ARRIVES"].str.split(":")
          df["FLIGHT_DEST_ARRIVES_HOURS"] = df["FLIGHT_DEST_ARRIVES"].str[0]
          df["FLIGHT_DEST_ARRIVES_MINS"] = df["FLIGHT_DEST_ARRIVES"].str[1]

In [16]:  df=df.drop(['FLIGHT_DEST_ARRIVES'], axis = 1)
```

Converting all object data types to numerical data types.



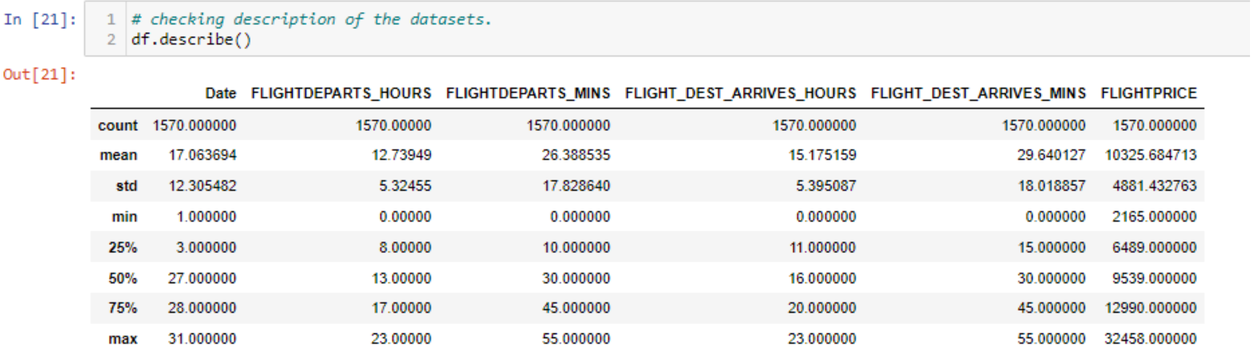
```
In [19]:  # converting all object data column to integer column.
          df['Date'] = df['Date'].astype("int")
          df['FLIGHTDEPARTS_HOURS'] = df['FLIGHTDEPARTS_HOURS'].astype("int")
          df['FLIGHTDEPARTS_MINS'] = df['FLIGHTDEPARTS_MINS'].astype("int")
          df['FLIGHT_DEST_ARRIVES_HOURS'] = df['FLIGHT_DEST_ARRIVES_HOURS'].astype("int")
          df['FLIGHT_DEST_ARRIVES_MINS'] = df['FLIGHT_DEST_ARRIVES_MINS'].astype("int")
          df['FLIGHTPRICE'] = df['FLIGHTPRICE'].astype("int")
```

Checking datatypes:



```
In [20]:  # checking datatypes of the dataframe
          df.dtypes

Out[20]:  AIRWAY                       object
          Date                          int32
          Month                        object
          FROMCITY                     object
          TOCITY                       object
          FLIGHTDEPARTS_HOURS           int32
          FLIGHTDEPARTS_MINS            int32
          FLIGHT_DEST_ARRIVES_HOURS     int32
          FLIGHT_DEST_ARRIVES_MINS      int32
          FLIGHTPRICE                   int32
          dtype: object
```

Checking description of the dataset:



In [21]:

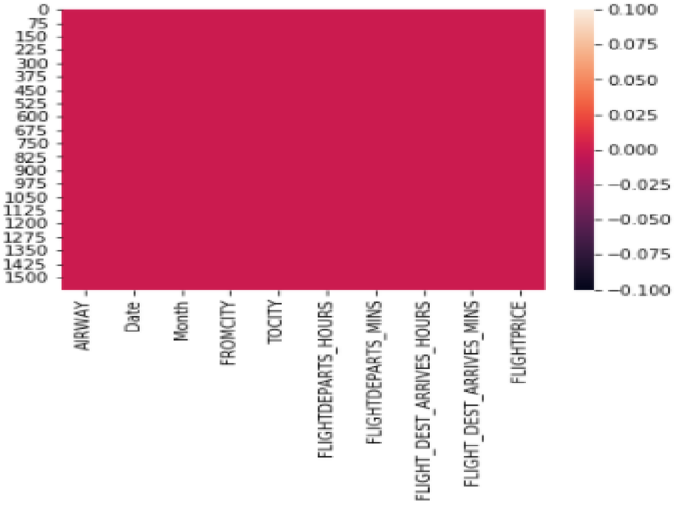
```
# checking description of the datasets.
df.describe()
```

Out[21]:

	Date	FLIGHTDEPARTS_HOURS	FLIGHTDEPARTS_MINS	FLIGHT_DEST_ARRIVES_HOURS	FLIGHT_DEST_ARRIVES_MINS	FLIGHTPRICE
count	1570.000000	1570.00000	1570.000000	1570.000000	1570.000000	1570.000000
mean	17.063694	12.73949	26.388535	15.175159	29.640127	10325.684713
std	12.305482	5.32455	17.828640	5.395087	18.018857	4881.432763
min	1.000000	0.00000	0.000000	0.000000	0.000000	2165.000000
25%	3.000000	8.00000	10.000000	11.000000	15.000000	6489.000000
50%	27.000000	13.00000	30.000000	16.000000	30.000000	9539.000000
75%	28.000000	17.00000	45.000000	20.000000	45.000000	12990.000000
max	31.000000	23.00000	55.000000	23.000000	55.000000	32458.000000

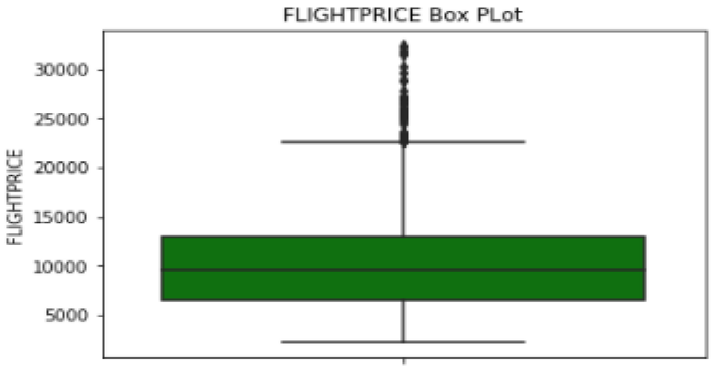
Observation: We can see that Flight price columns has mean value greater than the median which indicates the presence of Outliers in the dataset.

Checking Null values:



Observation: There are no Null values present in the dataset.

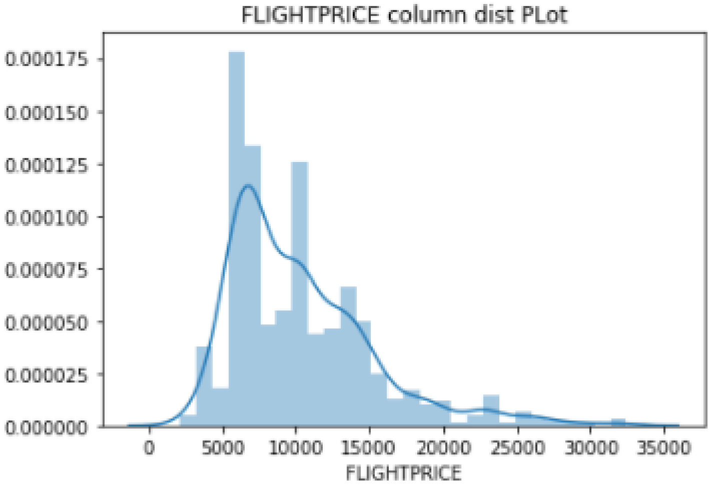
Checking outliers: using box plot of all numerical columns, we found out that only Flight Price column is having outliers in them.



Observation: we can see the presence of outliers in above box plot of flight price column.

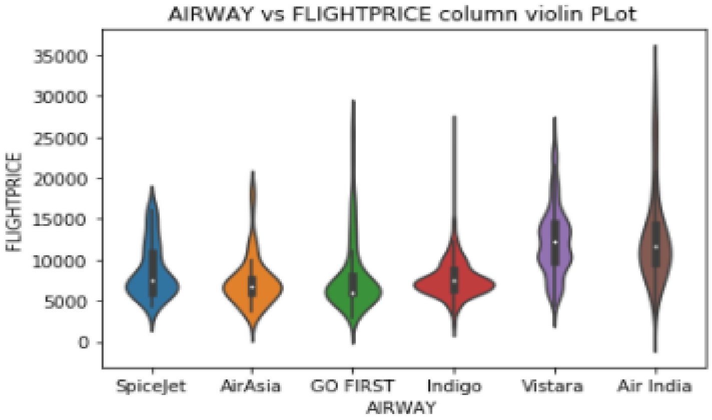
Checking skewness:

we checked for the skewness in almost all the column, and we found Flight price column as right sided skewed.



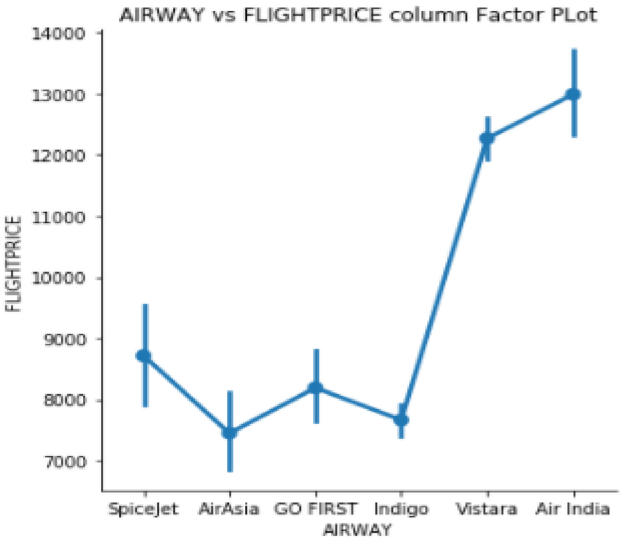
Observation: FLIGHTPRICE COLUMN IS Right sided skewed.

Violin Plot:



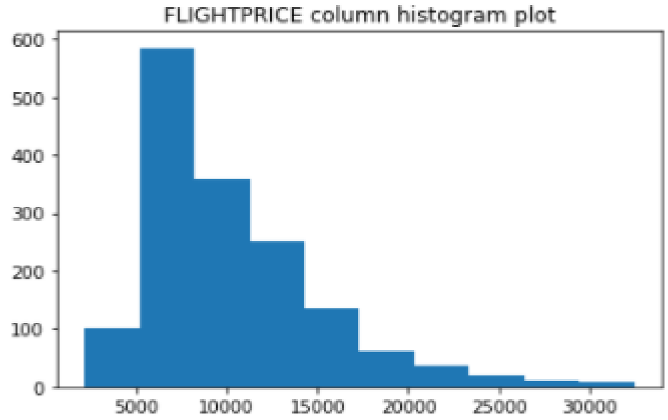
Observation: From the above violin plot, we can see that AirIndia is the most expensive flight, while spicejet is the cheapest flight among all other flights.

Factor Plot:



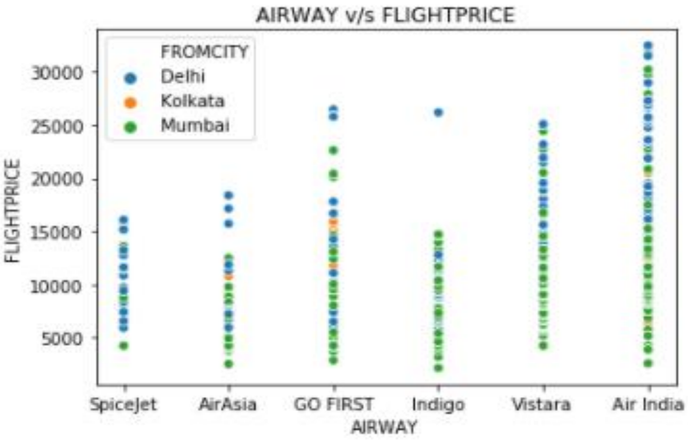
Observation: From the above Factor plot, we can easily make out that Air India is the most expensive flight while Air Asia, Indigo seems to be tbe cheapest flight among all other filghts.

Histogram Plot:



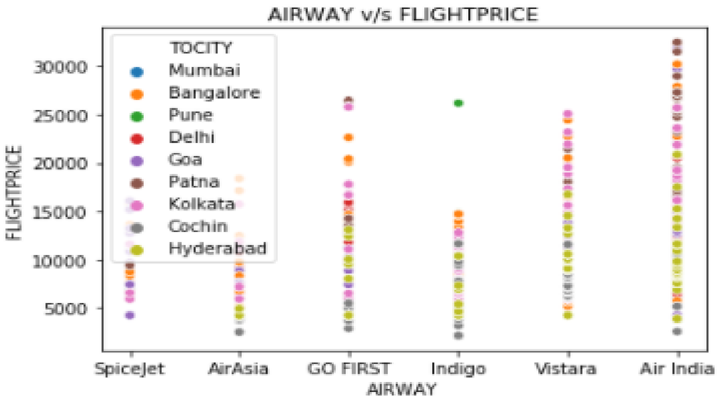
Observation: From the above histogram plot we can make out that, most of the Flightprices belongs to 5000 to 15000 price range while least flights are above 20000 price range.

Scatter Plot 1)



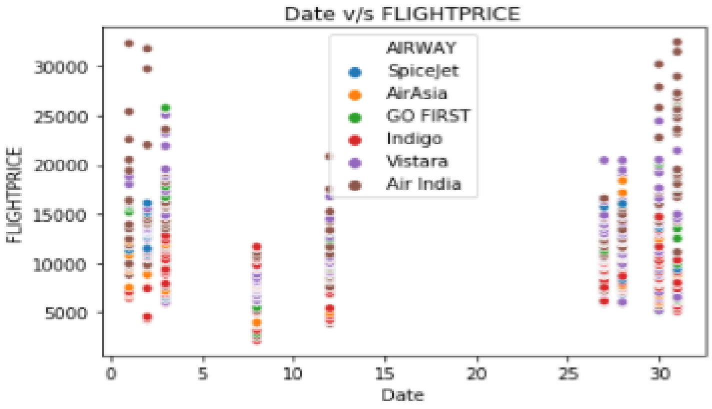
Observation: We can see that from the above scatter plot, AirIndia has maximum price range from less that 5000 to 35000 prices, while spicejet has least price range from 5000 to 18000 price range. Delhi city has maximum flights and most of the expensive flights are from Delhi only.

Scatter Plot 2):



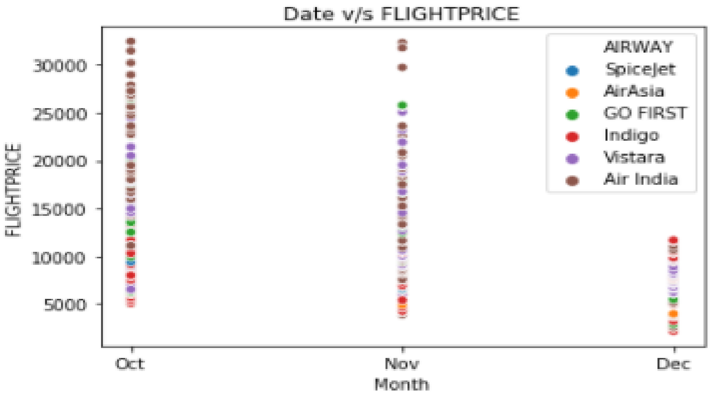
Observation: From the above scatter plot, we can make out that Air India seems to be having maximum flight price range where as Air Asia and Indigo having least flight price range.

Scatter Plot 3)



Observation: From the above scatter plot we can see that 30th of the month having maximum flight price range while 5th to 10th of the month having least flight price range.

Scatter Plot 4)



Observation: From the above scatter plot, we can make out that as we are checking flight price in october month, we can see that Oct seems to be having maximum flight price range while November less than October and December having least flight price range being in much future in callender.

LABEL ENCODING THE DATASET:

Before proceeding further, we need to confirm that all columns are numeric in nature, but we found out that there are columns which are of Object or Categorical data type in nature. We need to convert it into numeric data types, we can do that using Label Encoding technique for converting this column in to numeric. Label Encoding refers to converting the labels into numeric form so as to convert it into the machine readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. We Label Encode the categorical data columns like **'Airway', 'Month', 'FromCity', 'ToCity'** so that we can work on model building process smoothly.

REMOVING OUTLIERS USING Z-SCORE METHOD:

Here in this dataset, we have used z-score method of removing outliers. A z-score (also called a standard score) gives you an idea of how far from the mean a data point is. But more technically it's a measure of how many standard deviations below or above the population mean a raw score is. A z-score can be placed on a normal distribution curve. We can choose a threshold value, Here, in this problem we have choosen 3 as a threshold value above which all are outliers. After removing outliers using z-score method, we found that we are loosing almost **2.03 percent** of data, which is very much in the limits.

CORRELATION MATRIX: Correlation matrix is something by which we get to know how correlated all the variables with eachother. The correlation matrix is a matrix structure that helps the programmer analyze the relationship between the data variables. It represents the correlation value between a range of 0 and 1. The positive value represents good correlation and a negative value represents low correlation and value equivalent to zero(0) represents no dependency between the particular set of variables.

Observation:

From the above heatmap of correlation matrix, we can make out that, **Airway is having highly correlation with the Flight price with 0.15 units** while **FLIGHT_DEST_ARRIVES_MINS having 0.14 units** of correlation. so we can say that Airway and Flight_Dest_Arrives_mins column is having highly correlation with the flight price.

Creating X-Y Dataset: Basically we are splitting dataset into dependable variables target variables, here in this case, Y is the our target variable and X is our dependable variable.

Scaling the Dataset: The most important step in machine learning model making is to ensure all data columns are in same scale of values, we can bring them in same scale by scaling them using Standard Scaler scaling library. Standard Scaler removes the mean and scales each feature/variable to unit variance. This operation is performed feature-wise in an independent way. StandardScaler can be influenced by outliers (if they exist in the dataset, but we have already removed outliers) since it involves the estimation of the empirical mean and standard deviation of each feature.

Creating Train-Test Dataset: The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets Here we have kept 20% of dataset for testing the model while 80% of dataset for training the model.

MODEL MAKING PROCESS: Here in our problem we will try making machine learning model with six different types of algorithms i.e. Linear Regression, Lasso - Ridge Regularization method, ElasticNet Regularization method, Random Forest Regressor, Ada Boost Regressor, Support Vector Regressor, Decision Tree methods. We will try to find out each model's Accuracy score, Mean Absolute error, Mean Squared Error and Root Mean Squared Error as well. Accuracy score will give us the percentage accuracy of the model in predicting the test datasets. Mean Absolute Error is nothing but the magnitude of difference between the prediction of an observation and the true value of that observation. Mean Squared Error is nothing but the average of the square of the difference between the original values and the predicted values. Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

LINEAR REGRESSION MODEL: 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. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). The term linear model implies that the model is specified as a linear combination of features. Based on training data, the learning process computes one weight for each feature to form a model that can predict or estimate the target value. Our LinearRegression Model has shown accuracy score of **9%**

MEAN ABSOLUTE ERROR: 3244.258868160091

MEAN SQUARED ERROR: 16814886.007092822

ROOT MEAN SQUARED ERROR: 4100.595811231927

r2 Score of Linear Regression model: 0.06664082533561544

Cross Validation Score of the Linear Regression model: 0.0775490647894849

LASSO REGULARIZATION MODEL: In statistics and machine learning, lasso (least absolute shrinkage and selection operator) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

Our Lasso regression model has shown accuracy of **9%**

RIDGE REGULARIZATION MODEL: Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

Our Ridge Regression model has shown accuracy of **9%**

ELASTICNET REGULARIZATION MODEL METHOD: Elastic net is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions. Elastic Net is an extension of linear regression that adds regularization penalties to the loss function during training.

Our ElasticNet Regression model has shown accuracy of **9%**

MEAN ABSOLUTE ERROR: 3243.3464636099284

MEAN SQUARED ERROR: 16795313.49319747

ROOT MEAN SQUARED ERROR: 4098.20857121712

Cross validation score of ElasticNet Regularization method is **7%**

SUPPORT VECTOR REGRESSION METHOD:
SVR is built based on the concept of Support Vector Machine or SVM. It is one among the popular Machine Learning models that can be used in classification problems or assigning classes when the data is not linearly separable.

Our **linear, poly and rbf kernel** all types of SupportVector Model has **shown negative and very less positive accuracy score**.

RANDOM FOREST REGRESSION METHOD:
Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Our Random Forest Regression model has shown accuracy of **94.24%**

MEAN ABSOLUTE ERROR: 1541.7894942022265

MEAN SQUARED ERROR: 5468128.389939923

ROOT MEAN SQUARED ERROR: 2338.4029571354727

Cross validation score of our random forest regressor model is **63.38%**

ADA BOOST REGRESSION METHOD:
AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are reassigned to each instance, with higher weights assigned to incorrectly classified instances. An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction.

Our Ada Boost Regression model has shown accuracy of **38.71%**

MEAN ABSOLUTE ERROR: 2706.104778028869

MEAN SQUARED ERROR: 10793401.629246907

ROOT MEAN SQUARED ERROR: 3285.331281506769

Cross validation score of our Ada Boost Regressor model is **35.65%**

DECISION TREE REGRESSION METHOD:

Decision tree builds regression or classification models in the form of a tree structure. It **breaks down a dataset into smaller and smaller subsets** while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Our Decision tree regression model has shown accuracy of **99.72%**

MEAN ABSOLUTE ERROR: 1623.1444805194806

MEAN SQUARED ERROR: 9439028.975018037

ROOT MEAN SQUARED ERROR: 3072.3002742274457

Cross validation score of our Decision Tree Regressor model is **37.65%**

FINAL MODEL SELECTION: After checking all models accuracy score and cross validation score and also their errors, we found out that **Ada Boost Regression Model** seems to be the best model, as it has good accuracy score and least error comparatively. The difference between the accuracy score and cross validation score is also minimal. So we'll consider this Random Forest Regression Model model as our final model which is having accuracy of **38.71%**

Observation: we can see that both the y_test and predicted y_test datasets falls in near about one line in the distplots. we can say that our model is well made and have decent accuracy score.

SAVING THE MODEL: Our final model is saved in pkl format with 'Vaibhav_Flight_Price_Prediction_project_Model.pkl' file.

CONCLUSION:

1) **Air India** is the most expensive flight.
2) **Spice Jet , Air Asia, Indigo** seems to be cheapest flights.
3) More nearner the journey date maximum the flight price.
4) Farer the journey date least the flight price.
5) Most of the expensive flights are from **Delhi and then Mumbai**.
6) Least the Journey time, expensive the flight price.
7) **Ada Boost Regression model** seems to be the best fit model as its showing less overfitting, verified by cross validation score.
8) Ada Boost Regression model is showing accuracy of **38.71%.**

THANK YOU