NAME OF THE PROJECT

# Flight Price Prediction

Submitted by:

Ankita Srivastava

# ACKNOWLEDGMENT

We have scraped the data for over 1500 cars using Selenium script from 4 different websites from different locations around the country. The websites are as followed:

1…yatra.com,

2 .. skyscanner.com

3. Makemytrip.com

4. goibibo.com

I use this link to make my dataset. Makemytrip.com https://www.makemytrip.com/flights

I use sklearn to find data coding description

# INTRODUCTION

## 1))Business Problem Framing

Flight ticket prices can be something hard to guess. we have been provided with prices of flight tickets for various airlines between the months of March and June ,and between various cities, using which we aim to build a model which predicts the prices of the flights using various input features.

## 2))Conceptual Background of the Domain Problem

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)

- ## Review of Literature

The first thing that we can do when tackling a data science problem is getting an understanding of the dataset that you are working with. Key observations and trends in the data were noted down. All correlations within the variables and the output ‘*price*’ were monitored. For this you can use df.info,df.head()etc..

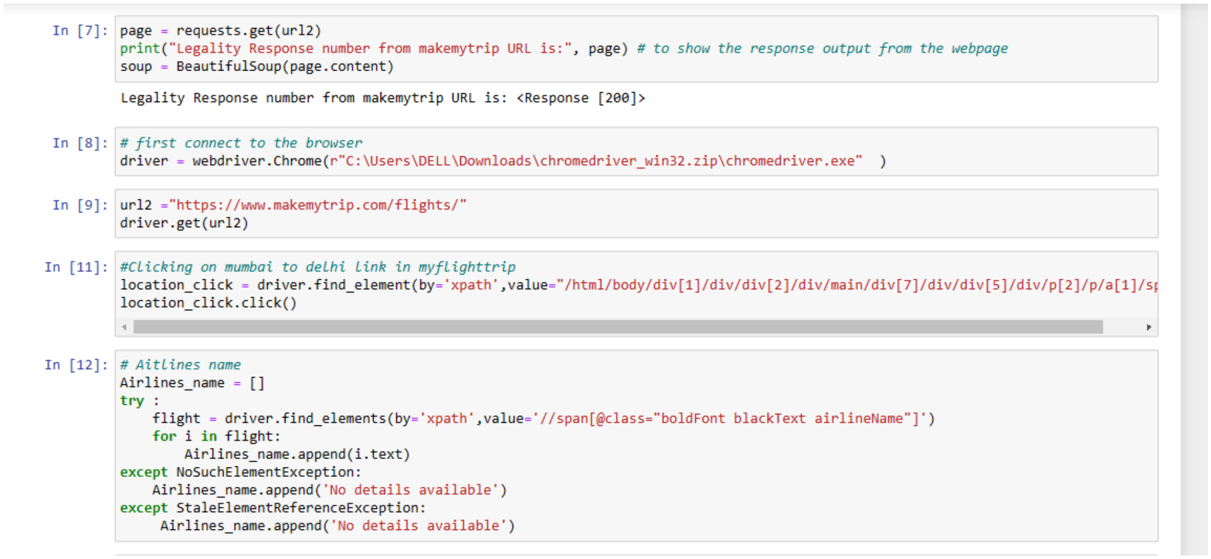
## Analytical Problem Framing

- ### Mathematical/ Analytical Modeling of the Problem

  Here matematical function use calculating the data loss. Data describe the stastistical function. Some regression model use to build the regression model.

- ### Data Sources and their formats

  We use some flight booking website to build the dataset.like I use makemytrip.com  to make my project.i used only flight booking detail.firstly I scrap all data using Beautiful soop method and after that I make a prediction model.where price is target.



```
In [7]: page = requests.get(url2)
        print("Legality Response number from makemytrip URL is:", page) # to show the response output from the webpage
        soup = BeautifulSoup(page.content)

        Legality Response number from makemytrip URL is: <Response [200]>

In [8]: # first connect to the browser
        driver = webdriver.Chrome(r"C:\Users\DELL\Downloads\chromedriver_win32.zip\chromedriver.exe"   )

In [9]: url2 ="https://www.makemytrip.com/flights/"
        driver.get(url2)

In [11]: #Clicking on mumbai to delhi link in myflighttrip
         location_click = driver.find_element(by='xpath',value="/html/body/div[1]/div/div[2]/div/main/div[7]/div/div[5]/div/p[2]/p/a[1]/sp
         location_click.click()

In [12]: # Aitlines name
         Airlines_name = []
         try :
             flight = driver.find_elements(by='xpath',value='//span[@class="boldFont blackText airlineName"]')
             for i in flight:
                 Airlines_name.append(i.text)
         except NoSuchElementException:
             Airlines_name.append('No details available')
         except StaleElementReferenceException:
              Airlines_name.append('No details available')
```

Scrap the data

- ### Data Preprocessing Done

  1.firstly I check the null values in this data, no any null values are present.
  2.how many unique values are present?
  3.any white spaces are present or not?
  4.which one is categorical data and which is continoues?
  5.which type of this problem regression or classification? When

we check all these thing I clean all data and fill appropriate. After that data visualization part proceed.

- ## Data Inputs- Logic- Output Relationships

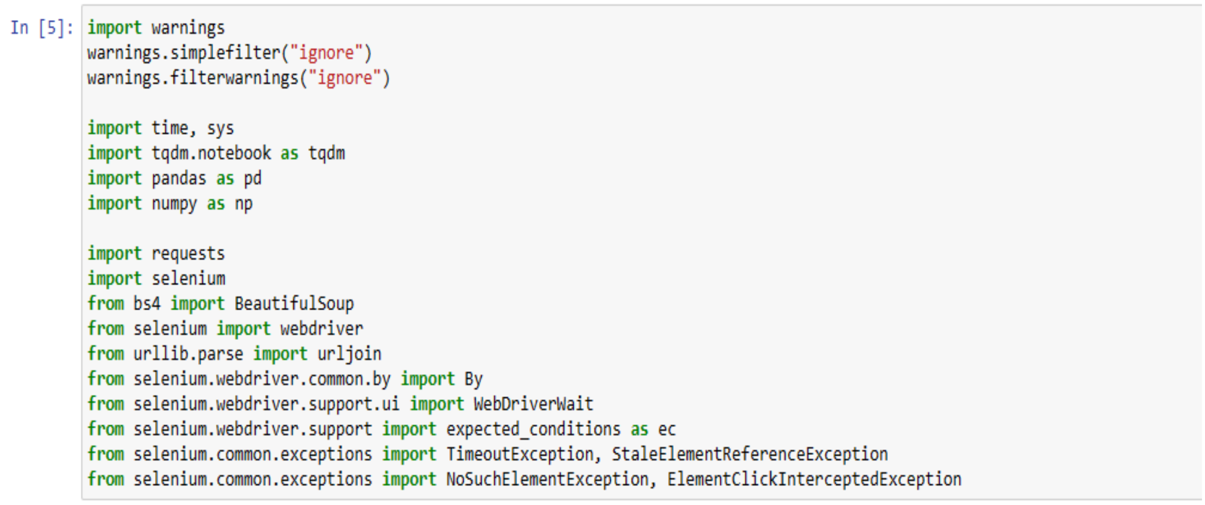
  Data model preparation depend on the large data . it has 228 rows and 7 column. every column affect the o/p. mostly depend on the duration, means how much it take time to one place from another.

- ## State the set of assumptions (if any) related to the problem under consideration

  I assume that target depend on flight name but I was wrong..it's totally depend on Duration.

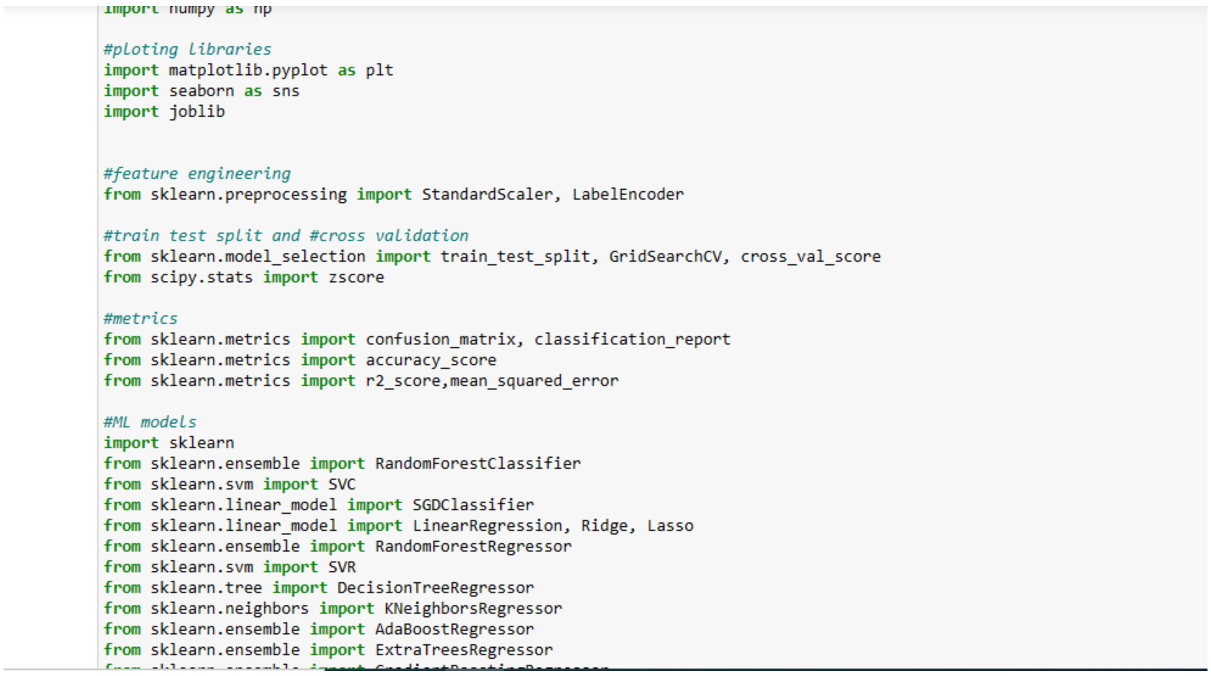
- ## Hardware and Software Requirements and Tools Used

  Data modelling need 8 Gb RAM . I can make model on Google colab or jupitor notebook. Here I use 2 type libraries one for scaraping the data and other for data modelling.



```
In [5]: import warnings
        warnings.simplefilter("ignore")
        warnings.filterwarnings("ignore")

        import time, sys
        import tqdm.notebook as tqdm
        import pandas as pd
        import numpy as np

        import requests
        import selenium
        from bs4 import BeautifulSoup
        from selenium import webdriver
        from urllib.parse import urljoin
        from selenium.webdriver.common.by import By
        from selenium.webdriver.support.ui import WebDriverWait
        from selenium.webdriver.support import expected_conditions as ec
        from selenium.common.exceptions import TimeoutException, StaleElementReferenceException
        from selenium.common.exceptions import NoSuchElementException, ElementClickInterceptedException
```

This is for scraping the data

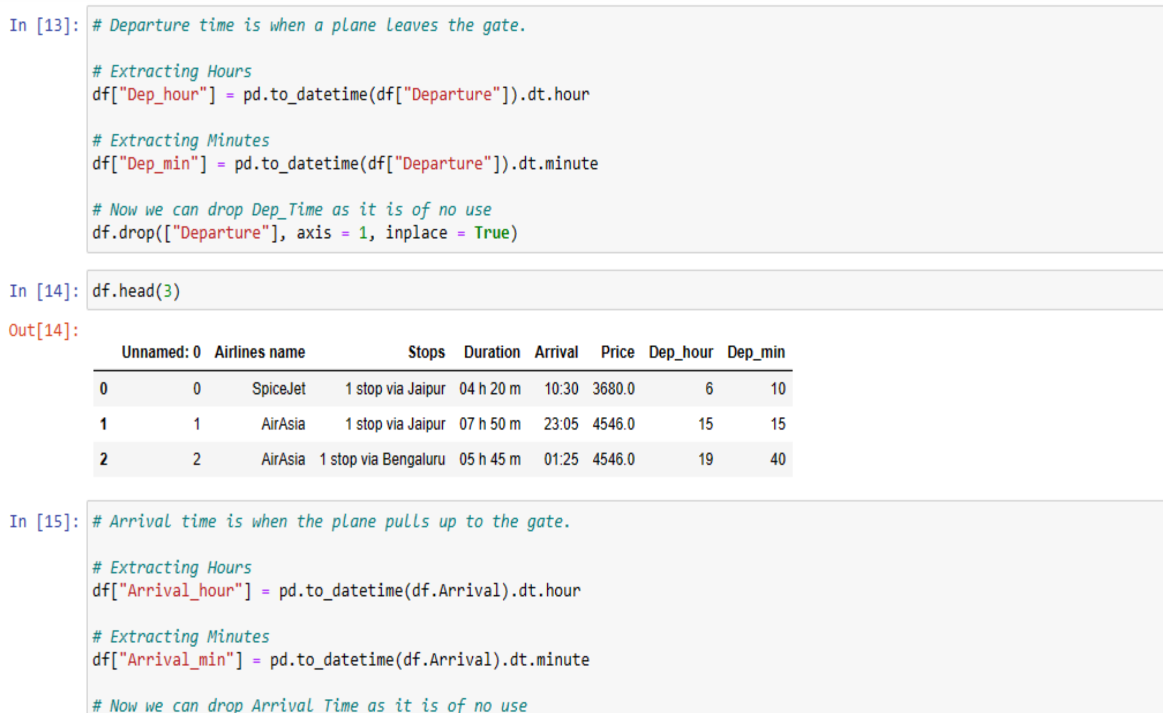


```python
import numpy as np

#ploting libraries
import matplotlib.pyplot as plt
import seaborn as sns
import joblib


#feature engineering
from sklearn.preprocessing import StandardScaler, LabelEncoder

#train test split and #cross validation
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from scipy.stats import zscore

#metrics
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score,mean_squared_error

#ML models
import sklearn
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import ExtraTreesRegressor
```

This is for model building

# Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)



```
In [13]: # Departure time is when a plane leaves the gate.

         # Extracting Hours
         df["Dep_hour"] = pd.to_datetime(df["Departure"]).dt.hour

         # Extracting Minutes
         df["Dep_min"] = pd.to_datetime(df["Departure"]).dt.minute

         # Now we can drop Dep_Time as it is of no use
         df.drop(["Departure"], axis = 1, inplace = True)
```

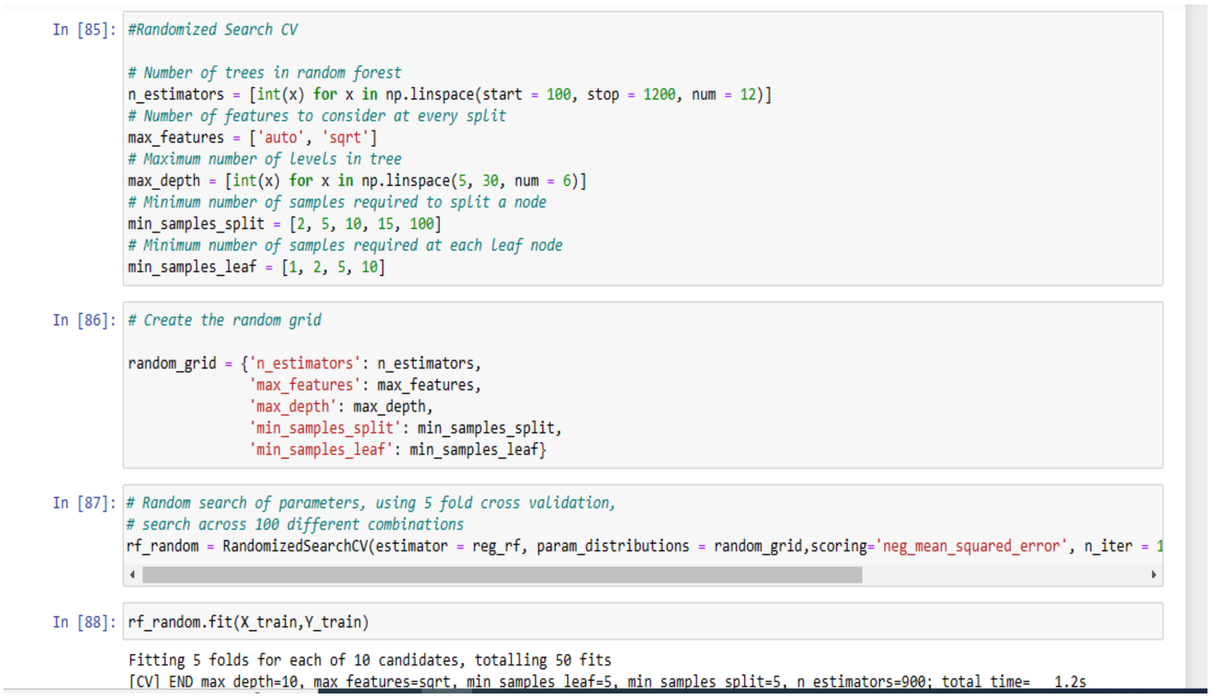
```
In [14]: df.head(3)
```

Out[14]:

| | Unnamed: 0 | Airlines name | Stops | Duration | Arrival | Price | Dep_hour | Dep_min |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | SpiceJet | 1 stop via Jaipur | 04 h 20 m | 10:30 | 3680.0 | 6 | 10 |
| **1** | 1 | AirAsia | 1 stop via Jaipur | 07 h 50 m | 23:05 | 4546.0 | 15 | 15 |
| **2** | 2 | AirAsia | 1 stop via Bengaluru | 05 h 45 m | 01:25 | 4546.0 | 19 | 40 |

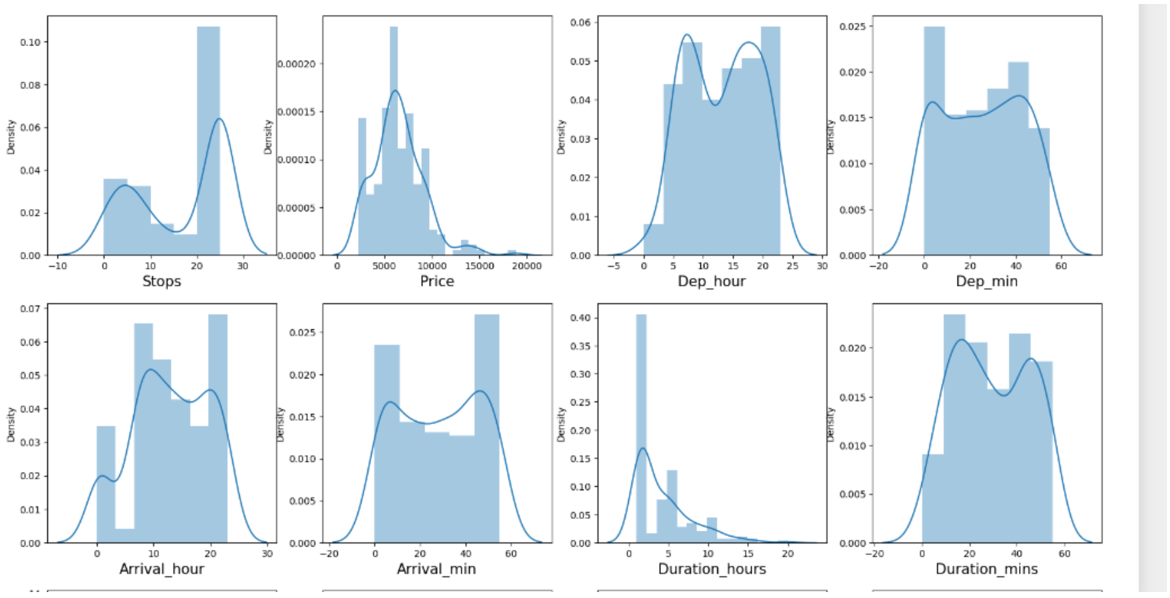
```
In [15]: # Arrival time is when the plane pulls up to the gate.

         # Extracting Hours
         df["Arrival_hour"] = pd.to_datetime(df.Arrival).dt.hour

         # Extracting Minutes
         df["Arrival_min"] = pd.to_datetime(df.Arrival).dt.minute

         # Now we can drop Arrival Time as it is of no use
```

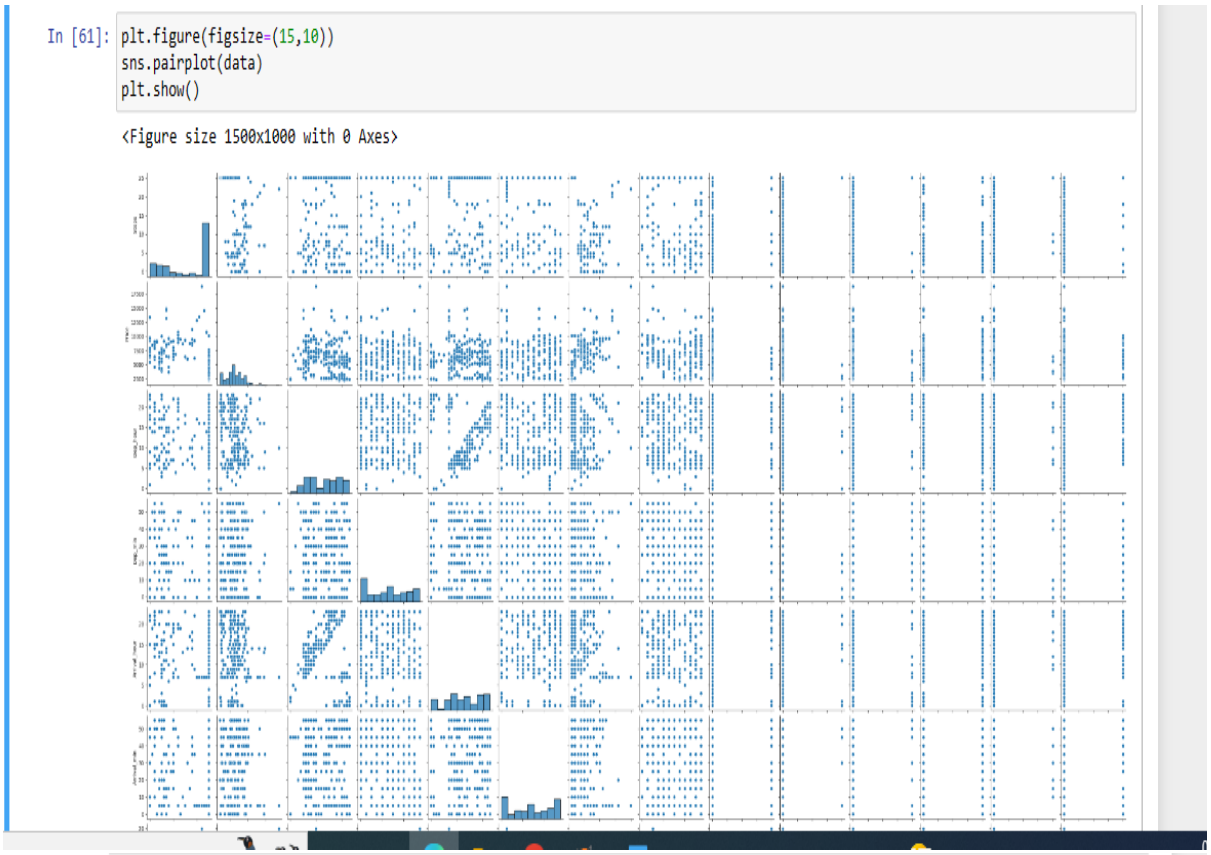
- Run and Evaluate selected models



```
In [85]: #Randomized Search CV

         # Number of trees in random forest
         n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
         # Number of features to consider at every split
         max_features = ['auto', 'sqrt']
         # Maximum number of levels in tree
         max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
         # Minimum number of samples required to split a node
         min_samples_split = [2, 5, 10, 15, 100]
         # Minimum number of samples required at each leaf node
         min_samples_leaf = [1, 2, 5, 10]

In [86]: # Create the random grid

         random_grid = {'n_estimators': n_estimators,
                        'max_features': max_features,
                        'max_depth': max_depth,
                        'min_samples_split': min_samples_split,
                        'min_samples_leaf': min_samples_leaf}

In [87]: # Random search of parameters, using 5 fold cross validation,
         # search across 100 different combinations
         rf_random = RandomizedSearchCV(estimator = reg_rf, param_distributions = random_grid,scoring='neg_mean_squared_error', n_iter = 1

In [88]: rf_random.fit(X_train,Y_train)

         Fitting 5 folds for each of 10 candidates, totalling 50 fits
         [CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.2s
```

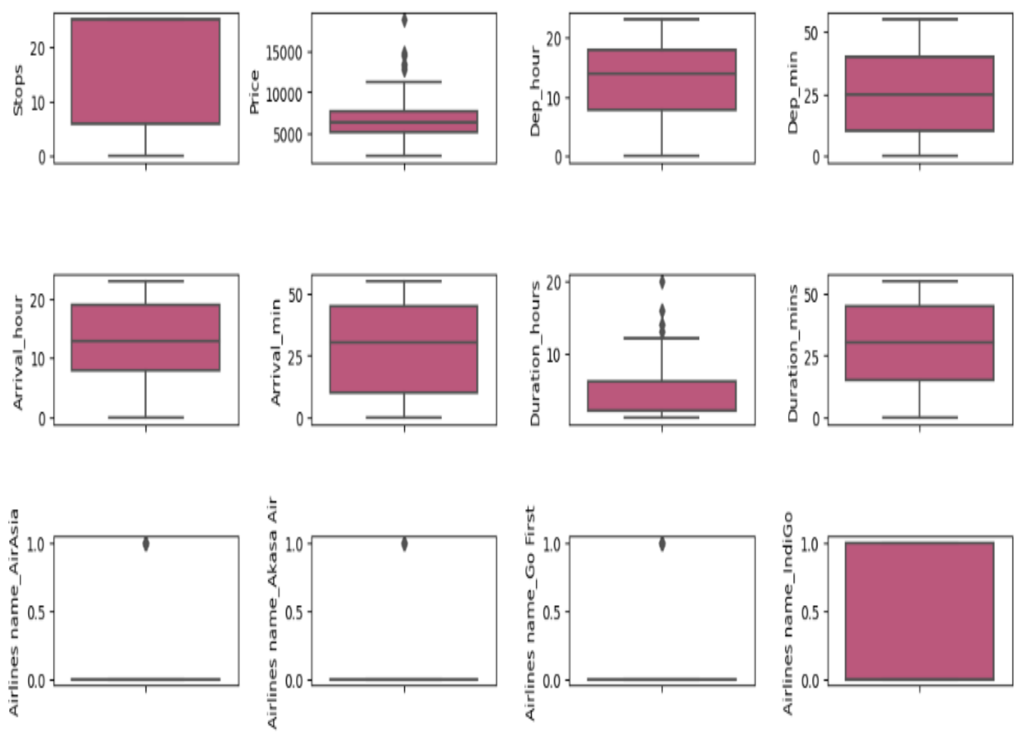
- Visualizations

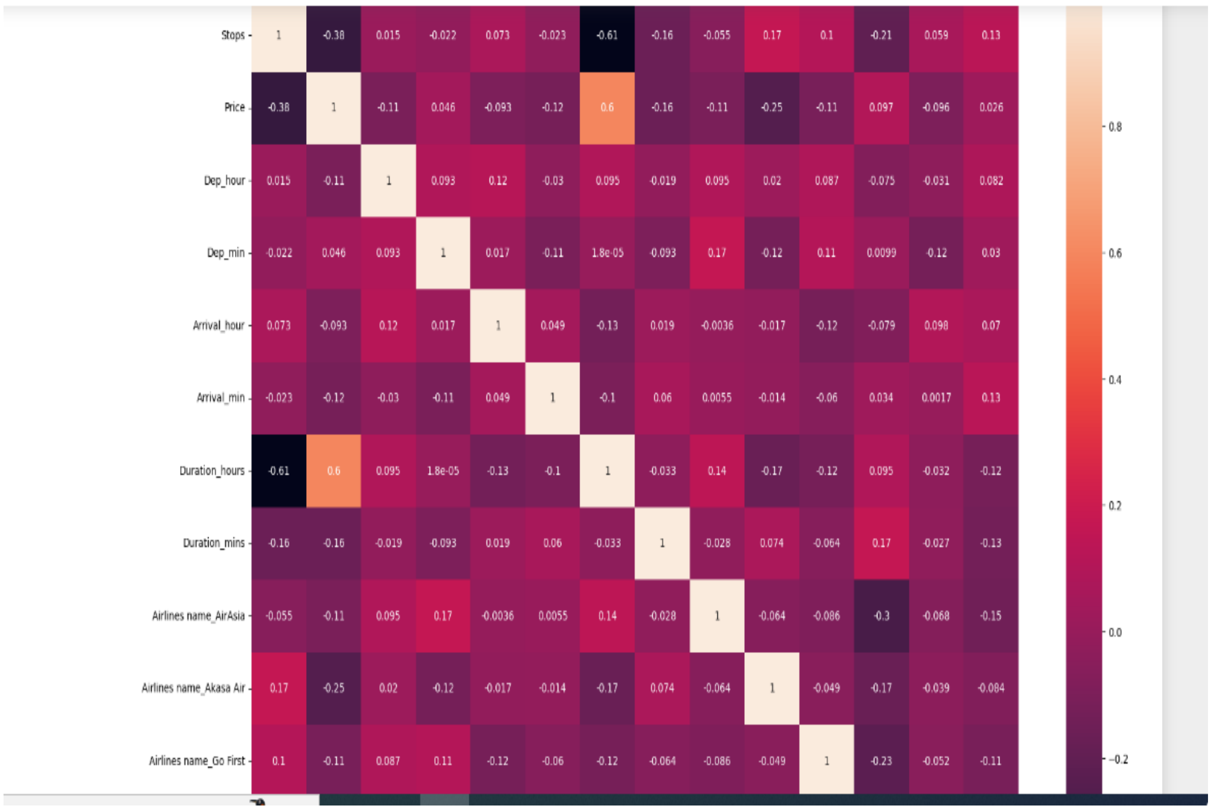




```
In [61]: plt.figure(figsize=(15,10))
         sns.pairplot(data)
         plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```





- ## Interpretation of the Results

I can see after data preprocessing step only duration hour column highly correlated with target column price.here minor outliers are present in continoues and categorical column. but i did not remove outliers from numerical column.but I have less data so I did not accept data after removing outliers because it can not build my train and test data..

## CONCLUSION

- ### Key Findings and Conclusions of the Study

  Here, we can see that all the predicted prices are either equal or nearly equal to the other airlines prices. Hence we conclude that our model 'price ' is working very well. And we shall save it for further use..

- ### Learning Outcomes of the Study in respect of Data Science

  1..use dist plot to show data distribution is normal.
  2...use line plot to show features are linear with target.
  3..using Heat map to show correlation b/w target and features.

  Random forest regressior is good perform other then all algorithm.but it also not showing satisfied result. First challenge was to scrap all data from different website. 2 nd was to create dataframe and convert it to string to float

- ### Limitations of this work and Scope for Future Work

As a part of future work, we aim at the variable choices over the algorithms that were used in the project. We could only explore two algorithms whereas many other algorithms exist and might be more accurate. More specifications will be added to a system or provide more accuracy in terms of price in the system