FLIGHT PRICE PREDICTION

Submitted by:

ASHISH CHAND

ACKNOWLEDGMENT

This flight data is collected from the online website of flight booking websites. The name of the website is Easemytrip.com. This project contains Excel file of scraped data, jupyter notebook for writing the code for the model training. In this project the data is collected with the help of selenium.

INTRODUCTION

- ## Business Problem Framing

  The flight ticket booking with less amount of risk and with cheap cost is what everyone wanted. 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. 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 in real world.

- ## Conceptual Background of the Domain Problem

  In today's world, as everything is tends to online. Like bookings of train, bus and the flights. So it's very useful for the people to get the whole details about the ticket by sitting in home or office rather than going to the station standing in queue for the booking. Therefore, it is very important to provide the correct information about the ticket online. In this project here I scrap the data of flight fares with other features/factors which affect the fares.

  To buy flight fares there are many online websites are there. Our task is to get the data from these online websites.

  So, this project is about to collect the data of flight fares with features which affect the cost online which contain Airline Name, Source from where the flight takes off, Destination where flight reached , Duration between the travelling , Number of stops between the source and destination, Departure date , departure time, Arrival time and the price. Basically everything which is important for the person to book a flight ticket.

  In this project, first I have to scrape the car data online and after that train the data for the prediction of upcoming dataset of flight fares. Also the data science part will be done like Exploratory Data

Analysis, data pre-processing, model building, model evaluation and selecting the best model.

- ## Review of Literature

  Ticket booking online is one of the most necessary and time-saving thing in today's world, people use online website and resources for the information and for booking the flight ticket. This project is about the get the cheapest available ticket on a given flight gets more and less expensive over time, our task is make a model to predict flight fares.

  So this project is done two phase - first is to collect the data websites of flights and collect all the important and the necessary details of the flight ticket booking. So that in the second phase, the machine learning model can be trained to predict the fare of ticket for the given flight, by giving it some data values.

  Data science comes as a very important tool to solve problems in the domain to help the client to increase their customers, profits, improving their marketing strategies and focusing on changing trends in ticket price data. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for flight price prediction. Our client are looking for new machine learning models from new data.

  So we need to build a model using Machine Learning in order to predict the flight ticket price value of the prospective data and give the ticket price output.

- Motivation for the Problem Undertaken

  Anyone who has booked a flight ticket knows how unexpectedly the prices vary. Airlines use using sophisticated quasi-academic tactics which they call "revenue management" or "yield management". 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. As a data scientist it is very interesting to know about all the information about the flights and the ticket booking according to fares, what are the factors which are necessary for booking the cheapest ticket for a given flight.
  Data science comes with a very important tool to solve the problems in the domain to help the make the model for flight price prediction and the give the good information regarding the factors which are necessary for getting the cheapest available ticket for a given flight.
  Recommendation systems are some of the machine learning techniques used for achieving the business goals for getting the cheapest available ticket websites.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

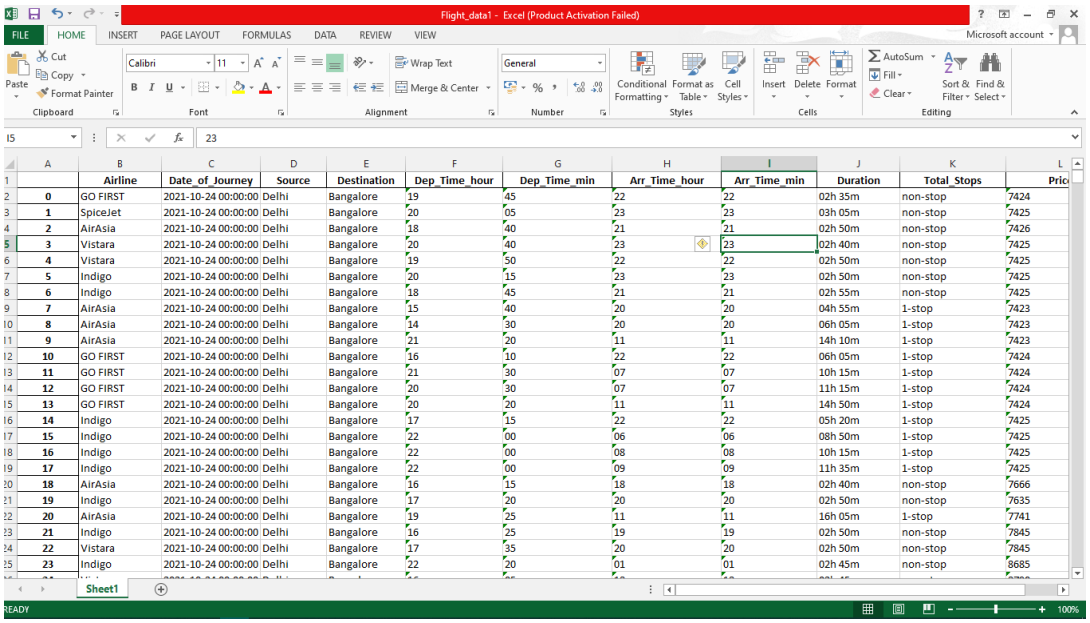
  In this project, as we analyse the dataset we get to know that:

  - The first step of this project is collect the data online from the online flight ticket booking websites.
  - After that I change the datatypes of the target column as it is object datatype which needs to be change into the float datatype
  - The departure date is split into departure day and departure month, similarly the duration time is split into hours and minutes and same done with the departure times and arrival times for the better understanding of the data.
  - To see the inter relation of features with the target column we use the correlation method, and then visualize with the help of Heatmap to see the correlation between features and with the target column.
  - Here I analyse that the Fare changes as duration of the travelling increases
  - The flight fares in small increment
  - Early morning flights are less expensive after that it increases.
  - And then I done the Feature selection process where I convert the categorical column into the numerical with the help of label encoding.
  - And finally the dataset get divided into x(without target column) and y(only target column present) for the model training.

  And at last the model with highest r2 score is selected for the model prediction.

- Data Sources and their formats

  The data is scrap from the online website for flight ticket book https://www.easemytrip.com/ with the help selenium and save it in the form of excel sheet.



	Airline	Date_of_Journey	Source	Destination	Dep_Time_hour	Dep_Time_min	Arr_Time_hour	Arr_Time_min	Duration	Total_Stops	Price
0	GO FIRST	2021-10-24 00:00:00	Delhi	Bangalore	19	45	22	22	02h 35m	non-stop	7424
1	SpiceJet	2021-10-24 00:00:00	Delhi	Bangalore	20	05	23	23	03h 05m	non-stop	7425
2	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	18	40	21	21	02h 50m	non-stop	7426
3	Vistara	2021-10-24 00:00:00	Delhi	Bangalore	20	40	23	23	02h 40m	non-stop	7425
4	Vistara	2021-10-24 00:00:00	Delhi	Bangalore	19	50	22	22	02h 50m	non-stop	7425
5	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	20	15	23	23	02h 50m	non-stop	7425
6	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	18	45	21	21	02h 55m	non-stop	7425
7	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	15	40	20	20	04h 55m	1-stop	7423
8	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	14	30	20	20	06h 05m	1-stop	7423
9	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	21	20	11	11	14h 10m	1-stop	7423
10	GO FIRST	2021-10-24 00:00:00	Delhi	Bangalore	16	10	22	22	06h 05m	1-stop	7424
11	GO FIRST	2021-10-24 00:00:00	Delhi	Bangalore	21	30	07	07	10h 15m	1-stop	7424
12	GO FIRST	2021-10-24 00:00:00	Delhi	Bangalore	20	30	07	07	11h 15m	1-stop	7424
13	GO FIRST	2021-10-24 00:00:00	Delhi	Bangalore	20	20	11	11	14h 50m	1-stop	7424
14	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	17	15	22	22	05h 20m	1-stop	7425
15	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	22	00	06	06	08h 50m	1-stop	7425
16	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	22	00	08	08	10h 15m	1-stop	7425
17	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	22	00	09	09	11h 35m	1-stop	7425
18	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	16	15	18	18	02h 40m	non-stop	7666
19	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	17	20	20	20	02h 50m	non-stop	7635
20	AirAsia	2021-10-24 00:00:00	Delhi	Bangalore	19	25	11	11	16h 05m	1-stop	7741
21	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	16	25	19	19	02h 50m	non-stop	7845
22	Vistara	2021-10-24 00:00:00	Delhi	Bangalore	17	35	20	20	02h 50m	non-stop	7845
23	Indigo	2021-10-24 00:00:00	Delhi	Bangalore	22	20	01	01	02h 45m	non-stop	8685

  - The dataset contains 11 columns and 8869 rows.

  - Airline: The name of the airline.

  - Date_of_Journey: The date of the journey

  - Source: The source from which the service begins.

  - Destination: The destination where the service ends.

  - Dep_Time_hr: The hour when the journey starts from the source.

  - Dep_Time_min: The minutes when the journey starts from the source.

  - Arr_Time_hr: Hour of arrival at the destination.

  - Arr_Time_min: The minutes when the flight arrival at the destination.

  - Duration: Total duration of the flight.

- Total_Stops: Total stops between the source and destination.
- Price: The price of the Flight ticket.

- Data Preprocessing Done

In the Data Pre-processing, at first I check the null values present in the dataset, there were no null values present. After that I done cleaning of the columns i.e. getting the desirable output in the column with the help of functions i.e. splitting the date to day and month with the date time function.

After that with the Duration column is split into two columns where one column contain duration hour and the other one contains minutes of the duration of travelling.

Then the price column is converted into the float datatype with the help of Astype function.

After that I drop the two unnecessary columns which were not useful for the next processes.

After that I use the Uni-Variate analysis and check the columns graphically. And then the bivariate analysis is done between the two columns. Then I perform the feature engineering performing the one hot encoding and convert the categorical features into the numerical features. And after that scaling is done on the numerical columns.

After that the data is split into two parts –

In first part the variable X store columns without dependent/target column and

And the second part the variable ‘y’ store only the target column and after that the data is split into train and testing and then the different model algorithms performs and then the we select best model according to the highest R2 score among all with hyperparameter tuning and then the model prediction is done.

- Data Inputs- Logic- Output Relationships

  By the visualization of the data we see the relationship between the features and also with the correlation tells about the all features related to the Target variable. By the correlation method we can see the relationship and effects of features on the target variable. And also we see the correlation between the features.

  By correlation we came to see that some features (Duration hour, Total stops) are highly correlated with the target column (Price). That means if the value of dependent feature increases, Price also increases.

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

  Here I scrap the data from 25th October 2021 to the 3rd November 2021. That's why it is very important to understand the data and the plotting in that way.

- Hardware and Software Requirements and Tools Used

  The Project is done on the Window 10, here I use the Software Anaconda platform (Python 3.8.5 64 bit) and the code is written on the Jupyter Notebook where I run different python libraries for the better understanding. The libraries were:

  - Pandas – Pandas library is used for the uploading the file and then to create the dataframe from that file and manipulation of dataframe and use some in-built operations.

  - Numpy – Numpy is used for the mathematical operations in the dataframe. Here Numpy is used to calculate the mean, median, mode and also the various mathematical operations

were used in the dataset, which helps in the better understanding of the different features.

- Matplotlib – Matplotlib.pyplot library is used for the visualization for the dataset. And it helps to understand the data with help of graphs. And also it helps to label the graph, x-axis or y-axis. It also helps to select the size of the graph.

- Seaborn – Seaborn again is used for the visualization and it helps us to visualize the data more efficiently and clearly and understand it properly. Here I used the Heatmap which helps to understand to see the correlation between features and also here I use the various graphs like histogram, Bar graph, Scatterplots, line plots etc.

- Sci-kit learn – Sci-kit learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy. Scikit-learn library is very important in the machine learning process. Here I use the library for model selection such as Linear regression , Lasso, Ridge Support vector machine,k-nearest neighbour, some ensemble techniques such as RandomForest Regressor, metrices like Mean absolute error, root mean squared error , r2score, Pre-processing tools like Onehot encoder, StandardScaler and model selection like GridsearchCV , train test split and cross-validation value.

Importing the libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso,Ridge
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score,GridSearchCV
from sklearn.metrics import r2_score

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

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Here we first check the data is supervised or unsupervised by checking the target column and then when we see the output it shows it's a regression problem.

  - First, I import the necessary libraries, such as pandas, Numpy, Seaborn, Matplotlib, scipy and sklearn.
  - After that with the help of pandas I create the dataframe from the given set of data.
  - Then I check the information the dataframe which shows that 8869 rows and 11 features and also the data type datetime64 [ns] (1), int64 (6), object (5) were present.
  - Here some columns were present as the object data type which was converted into float and integer later.
  - Then I check the missing values in the dataframe and there were no null present in the dataframe.
  - Then I check the correlation of features and found that Duration hours, and Total number of stops are positively highly correlated first.
  - As our dataset is containing both Object data type and Integer, float data type.
  - As some columns were in object datatype, therefore with the help of astype function and lambda or map function we first capture the numerical value and then store them as a numeric value with float data type.
  - Then the duration time is split into two columns where one column contain duration hours while other contain minutes.
  - Similarly number of stores are stored as 0, 1 and 2, after that they converted into the integer datatype.
  - Bar plot were used to represent the discrete features for the better understanding of dataset.
  - Countplot were used to represent the count the values of categorical features

- Then I use the label Encoder and convert the categorical columns into the numerical columns.
- And then split the dataset in x and y where x contains all the dependent columns and y contains only target column.
- After that the model training is done and select the best model which have best R2 score and also the hyperparameter tuning is also the and after that the model is used for the prediction.

Random forest regressor is then used for the testing and for the prediction.

- Testing of Identified Approaches (Algorithms)

Here we use the following Algorithms used for training & Testing:

1. Linear Regression
2. Lasso
3. Ridge
4. Support vector machine
5. K-nearest Neighbors Regressor
6. Random forest Regressor

- Run and Evaluate selected models

The algorithms used were:

Model Training

```
In [98]: def evaluate(model, X_train, y_train, X_test, y_test):
             print('TRAIN')
             pred = model.predict(X_train)
             print(f'MEAN ABSOLUTE ERROR: {mean_absolute_error(y_train, pred)}')
             print(f'MEAN SQUARED ERROR: {mean_squared_error(y_train, pred)}')
             print(f'ROOT MEAN SQUARED ERROR: {np.sqrt(mean_squared_error(y_train, pred))}')
             print(f'R2 SCORE: {r2_score(y_train, pred)}')
             print('##############################')
             print('TEST')
             pred = model.predict(X_test)
             print(f'MEAN ABSOLUTE ERROR: {mean_absolute_error(y_test, pred)}')
             print(f'MEAN SQUARED ERROR: {mean_squared_error(y_test, pred)}')
             print(f'ROOT MEAN SQUARED ERROR: {np.sqrt(mean_squared_error(y_test, pred))}')
             print(f'R2 SCORE: {r2_score(y_test, pred)}')
```

Above is the function used and after that the model were run on it.

1. **Linear Regression :**

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

Linear Regression

```
[73]: model = LinearRegression()
      model.fit(X_train,y_train)

t[73]: LinearRegression()

[74]: evaluate(model, X_train, y_train, X_test, y_test)

      TRAIN
      MEAN ABSOLUTE ERROR: 2679.9415980927865
      MEAN SQUARED ERROR: 14028324.931251349
      ROOT MEAN SQUARED ERROR: 3745.4405523584737
      R2 SCORE: 0.287014185332396
      ##############################
      TEST
      MEAN ABSOLUTE ERROR: 2802.8569526060246
      MEAN SQUARED ERROR: 15694469.611196134
      ROOT MEAN SQUARED ERROR: 3961.624617653234
      R2 SCORE: 0.2822911111634766
```

2. **Lasso :**

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean.
The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.
Lasso Regression uses L1 regularization.
It is used when we have more number of features because it automatically performs feature selection.

Lasso

```
75]: param_grid = {
         'alpha': [0.1, 0.01, 0.0001, 0.002, 0.00105, 0.000001]
     }
     lasso = Lasso()
     lasso_grid = GridSearchCV(lasso, param_grid=param_grid,
                               scoring='neg_mean_squared_error',cv=10).fit(X_train, y_train)
     evaluate(lasso_grid, X_train, y_train, X_test, y_test)
```

```
TRAIN
MEAN ABSOLUTE ERROR: 2679.937005492601
MEAN SQUARED ERROR: 14028324.948954837
ROOT MEAN SQUARED ERROR: 3745.4405547218125
R2 SCORE: 0.28701418443262106
##############################
TEST
MEAN ABSOLUTE ERROR: 2802.8567031222433
MEAN SQUARED ERROR: 15694527.628947483
ROOT MEAN SQUARED ERROR: 3961.631940116028
R2 SCORE: 0.282288458008774
```

3. **Ridge:** 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.

Ridge

```
76]: ridge = Ridge()
     param_grid = {
         'alpha': [12, 12.1, 12.2, 12.3, 11.9, 11.8],
     }
     ridge_grid = GridSearchCV(ridge, param_grid=param_grid,
                               scoring='neg_mean_squared_error',
                               cv=10).fit(X_train, y_train)
     evaluate(ridge_grid, X_train, y_train, X_test, y_test)
```

```
TRAIN
MEAN ABSOLUTE ERROR: 2677.709431345811
MEAN SQUARED ERROR: 14035616.604166767
ROOT MEAN SQUARED ERROR: 3746.4138324759006
R2 SCORE: 0.28664358803162304
##############################
TEST
MEAN ABSOLUTE ERROR: 2803.4146663992574
MEAN SQUARED ERROR: 15737808.34722339
ROOT MEAN SQUARED ERROR: 3967.090665364656
R2 SCORE: 0.2803092285737322
```

4. **Support Vector machine**

   Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to

find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

SVR

```
']: svr=SVR()
param_grid={'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}

svr_grid = GridSearchCV(svr, param_grid=param_grid,
                           scoring='neg_mean_squared_error',
                           cv=10).fit(X_train, y_train)
evaluate(svr_grid, X_train, y_train, X_test, y_test)
```

```
TRAIN
MEAN ABSOLUTE ERROR: 1816.440440999374
MEAN SQUARED ERROR: 10382280.967570951
ROOT MEAN SQUARED ERROR: 3222.154708820008
R2 SCORE: 0.4723233821537076
##############################
TEST
MEAN ABSOLUTE ERROR: 2069.361750585756
MEAN SQUARED ERROR: 12754689.586279875
ROOT MEAN SQUARED ERROR: 3571.3708273266548
R2 SCORE: 0.41672740033895084
```

5. K- Nearest Neighbors regression

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood.

KNeighborsRegressor

```
78]: model=KNeighborsRegressor()
     model.fit(X_train,y_train)
     evaluate(model, X_train, y_train, X_test, y_test)
```

```
TRAIN
MEAN ABSOLUTE ERROR: 1562.1705998681607
MEAN SQUARED ERROR: 6302100.001674358
ROOT MEAN SQUARED ERROR: 2510.398375093953
R2 SCORE: 0.6796974745145363
##################################
TEST
MEAN ABSOLUTE ERROR: 2064.165936178393
MEAN SQUARED ERROR: 11228354.975809304
ROOT MEAN SQUARED ERROR: 3350.873763036934
R2 SCORE: 0.48652675924765054
```

6. **Random Forest Regressor**:

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

```
79]: Rfr = RandomForestRegressor(random_state = 0)
     param_grid = {  'bootstrap': [True],
                  'max_depth': [5, 10, None],
                  'max_features': ['auto', 'log2'],
                  'n_estimators': [5, 6, 7, 8, 9, 10, 11, 12, 13, 15]
                 }

     random_grid= GridSearchCV(Rfr, param_grid=param_grid, scoring='neg_mean_squared_error',
                               cv=10).fit(X_train, y_train)
     evaluate(random_grid, X_train, y_train, X_test, y_test)
```

```
TRAIN
MEAN ABSOLUTE ERROR: 575.1712628807446
MEAN SQUARED ERROR: 1074784.0436114403
ROOT MEAN SQUARED ERROR: 1036.717919017242
R2 SCORE: 0.9453743921186969
##################################
TEST
MEAN ABSOLUTE ERROR: 1525.928537935145
MEAN SQUARED ERROR: 7205409.602276727
ROOT MEAN SQUARED ERROR: 2684.289403599531
R2 SCORE: 0.6704962545804748
```

- Key Metrics for success in solving problem under consideration:

  The key metrices used along with the project are:

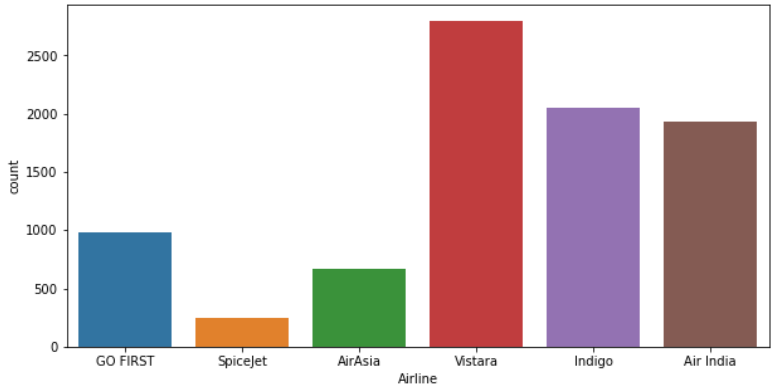
  - **Mean absolute error: (MAE)** represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set.
  - **Mean squared error**: (**MSE**) represents the difference between the original and predicted values extracted by squared the average difference over the data set.
  - **Root Mean squared error**: (**RMSE**) is the error rate by the square root of MSE.
  - **R2 score: R-squared (Coefficient of determination)** represents the coefficient of how well the values fit compared to the original values. The value from 0 to 1 interpreted as percentages. The higher the value is, the better the model is.

- **Visualizations**

  Here we use the following graph:

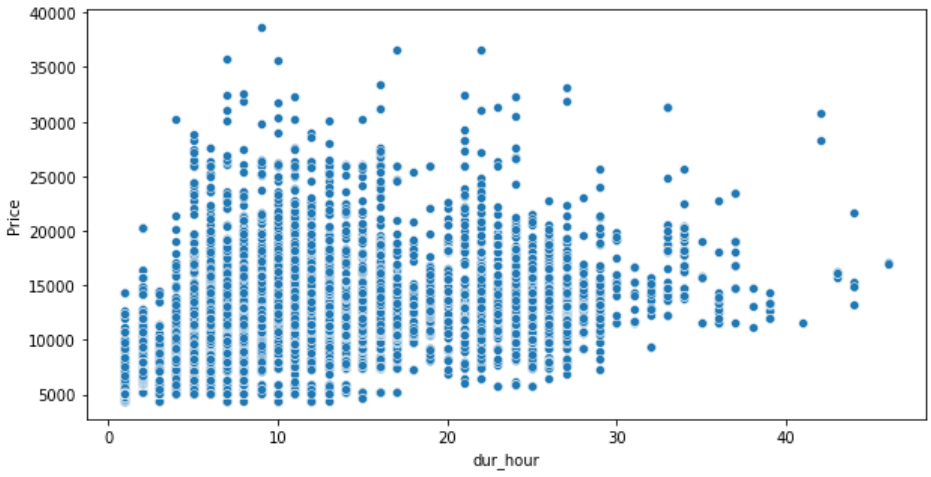
  Countplot: The countplot is majorly used for showing the observational count in different category based bins with the help of bars.

```
29]: #Lets count the different Airplanes data we have
plt.figure(figsize=(10,5))
sns.countplot(x="Airline",data=df)
plt.show()
```



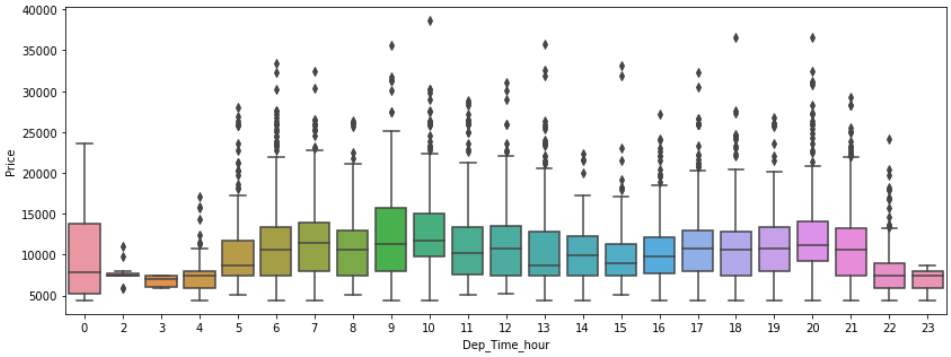
2. **Scatterplot** to see the distribution of data among two columns. Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them. The scatter() method in the matplotlib library is used to draw a scatter plot. Scatter plots are widely used to represent relation among variables and how change in one affects the other.

```
]: plt.figure(figsize=(10,5))
sns.scatterplot(x=df['dur_hour'],y=df['Price'])
plt.show()
```



3. **Box plot** : A Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

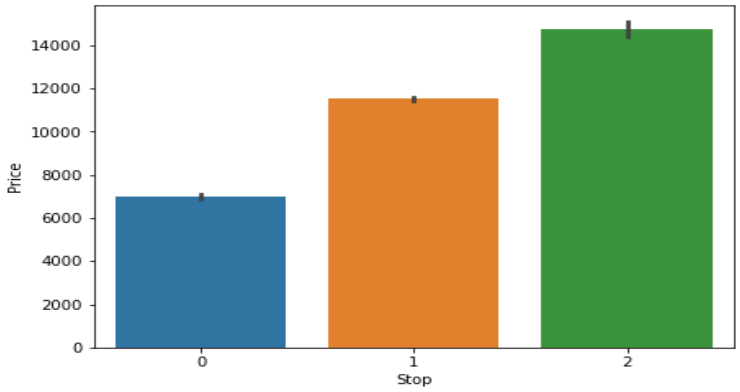
```
]: plt.figure(figsize=(14,5))
   sns.boxplot(x=df['Dep_Time_hour'],y=df['Price'])
   plt.show()
```



4. **Bar Plot :**

   A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories**.**

```
:  #Lets check from which stops and averge price of the flight was more
   plt.figure(figsize=(7,5))
   sns.barplot(x="Stop",y="Price",data=df)
   plt.show()
```

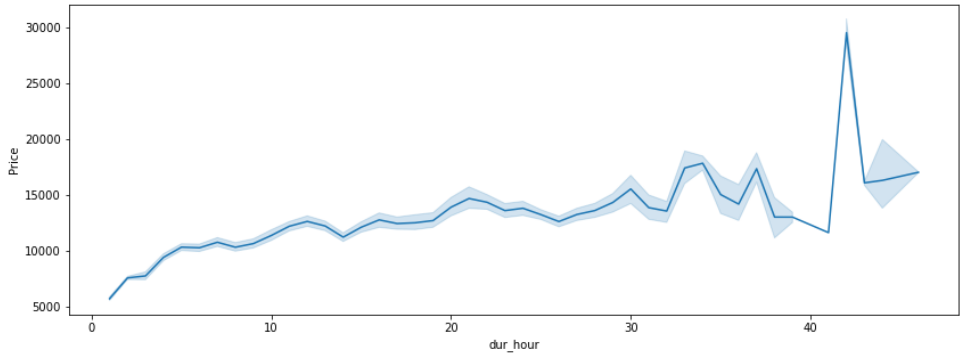


5. **Line plot:** A line graph—also known as a line plot or a line chart—is a graph that uses lines to connect individual data points. A line graph displays quantitative values over a specified

time interval.

```
8]: plt.figure(figsize=(14,5))
    sns.lineplot(x=df['dur_hour'],y=df['Price'])
```

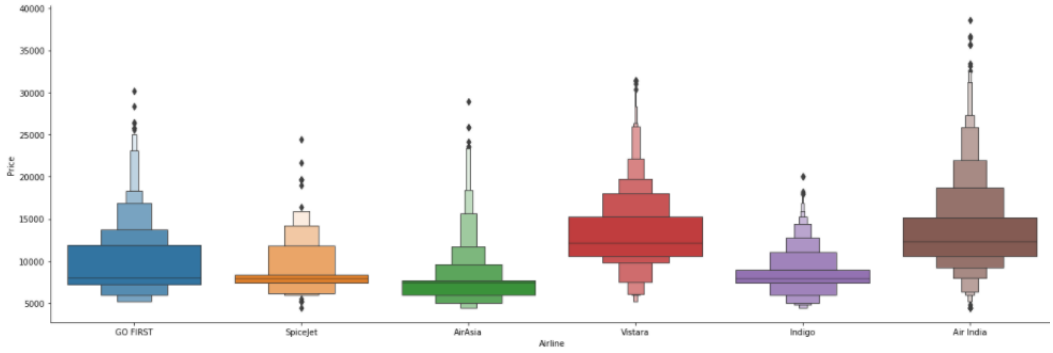
```
8]: <AxesSubplot:xlabel='dur_hour', ylabel='Price'>
```



6. **Catplot:** Catplot is a relatively new addition to Seaborn that simplifies plotting that involves categorical variables.

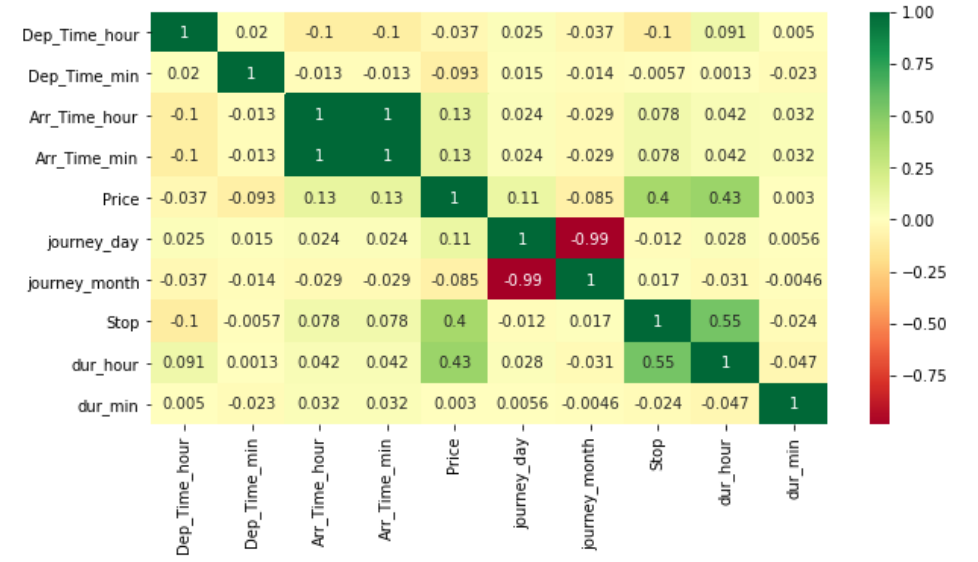
```
]: #Lets check which airline is expensive based on average price
   plt.figure(figsize=(10,10))
   sns.catplot(x="Airline",y="Price",data=df,kind='boxen',height=6,aspect=3)
   plt.show()
```

```
<Figure size 720x720 with 0 Axes>
```



7. **Heatmap** is used to see the correlation of columns.

```
#Lets see the graphical representation of the correlation
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(10,5))
#plot heat map
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



- ## Interpretation of the Results

This data is scraped online from the flight ticket website. Through the help of Heatmap we see that there are no null values are present in the dataset. As the dataset was not in the proper data type, first I convert the object datatype numerical column to the Integer and Float data type columns. And the I drop two columns which were not so important. And to check distribution in each columns I use**, count plot** for the categorical columns.

And then to check the relationship between the Target column i.e. price and other columns I used the **scatter plots, bar plots, boxplots and line plot** for the better understanding and meaningfulness.

And then I did the Feature selection process where I convert the categorical column into the numerical with the help of one hot encoding and then to bring all the numerical column into same scale, I use the standard scale method on the numerical columns except the target column and categorical columns.

And finally the dataset get divided into two variable x (All columns except target column) and y (only target column present) for the model training.

CONCLUSION

- Key Findings and Conclusions of the Study

  In this project prediction of flight price we find out that the Price is dependent with the -

  1. Duration (Total duration of the flight)
  2. Number of stops between the Source and the Destination

  That Means if the duration of the flight is more its ticket price will be high, i.e. the longer the flight duration the higher will be the prices.

  The price increases with number of stops increases.

  Early morning flights are less expensive after that the fare increases.

  Air India and Vistara airlines planes have the high price ticket with respect any other airlines.

  In this project we use different models like Linear Regression, Lasso, Ridge, SVR, KNN Regression and Random Forest Regression.

  Here we select the RandomForestRegressor model for our final model training and testing as its R2 score is highest among the all models we choose.

- Learning Outcomes of the Study in respect of Data Science

  In this project, I came to know about the flight ticket fares and the features which affect the fare and with the help of the visualization tool like Matplotlib and Seaborn, it get easier for the understanding.

  The study shows different-different regression algorithms when predicting Flight prices in India .The results were good for the data due to it being with features and having correlation.

  Hence, the data needs more features to be added preferably with a strong correlation with the flight ticket price. However, Random Forest gave the best R2 score overall. The final results of this study showed that Random Forest makes better prediction compared to other used algorithms.

  - *Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time?*
    - The airfares changes frequently as they want to get maximize profits and fill more seats as soon as possible.
    - It depends on the duration of the travelling if it is small it is less while the duration is more the price take large jump.
    - The Airfares tend to go up and down to maximize the profits and fill more seats and also it go up and down with respect to number of stops and the duration of travel.

  - *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?*
    - The best time for the customer to buy is to book as early as possible of travelling date because price

increases or remain constant as we get near to departure date so it's better to book early before the departure date i.e. 15-30 days early to get least risk and good return.

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

Future work on this study could be divided into five main areas to improve the result even further. Which can be done by:

- The used pre-processing methods do help in the prediction R2score. However, experimenting with different combinations of pre-processing methods to achieve less Root Mean squared error and Good R2 score.
- Make use of the available features and if they could be combined as binning features has shown that the data got improved.
- Training the datasets with different regression methods such as XGBoost and Catboost. In order to expand the comparison and check the performance.
- The correlation has shown the association in the data. Thus, attempting to enhance the data is required to make rich with features that vary and can provide a strong correlation relationship.
- Collected and analysed data for 16 routes which spanned across business & tourist routes in India.
- Collected more data as possible for better process.
- Some of the routes had non-decreasing prices and thus the model suggested to buy the ticket always.
- Implemented Gradient Boosting and more statistical analysis
- Added a Boolean parameter of Holidays.